NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

S6844

DRAFTSMANS DISPLAYS FOR CONTINGENCY TABLES USING A FULL-SCREEN, SCROLLABLE APL2 SPREADSHEET INPUT/OUTPUT EDITOR WITH APPLICATION TO THE PERSEREC DATABASE OF SPECIAL BACKGROUND INVESTIGATION

by

Uwe H. Steinfeld

March 1990

Thesis Advisor: Peter A. W. Lewis

Approved for public release; distribution is unlimited.

Unclassified

Security Classification of this Page

REPORT DOCUMENTATION PAGE

1a. Report Security Classification Unclassified	1b. Restrictive Markings
2a. Security Classification Authority	3. Distribution/Availability of Report Approved for public release; distribution is unlimited.
2b. Declassification/Downgrading Schedule	
4. Performing Organization Report Number(s)	5. Monitoring Organization Report Number(s)

6a. Name of Performing Organization Naval Postgraduate School	6b. Office Symbol 55	7a. Name of Monitoring Organization Naval Postgraduate School
6c. Address (*City, State, and ZIP code*) Monterey, CA 93943-5000		7b. Address (*City, State, and ZIP code*) Monterey, CA 93943-5000
8a. Name of Funding/Sponsoring Organization	8b. Office Symbol	9. Procurement Instrument Identification Number
8c. Address (*City, State, and ZIP code*)		10. Source of Funding Numbers

Program Element No	Project No	Task No	Work Unit Accession No

11. Title (*Include Security Classification*)

DRAFTSMANS DISPLAYS FOR CONTINGENCY TABLES USING A FULL-SCREEN, SCROLLABLE APL2 SPREADSHEET INPUT/OUTPUT EDITOR WITH APPLICATION TO THE PERSEREC DATABASE OF SPECIAL BACKGROUND INVESTIGATION

12. Personal Author(s) Uwe H. Steinfeld

13a. Type of Report Master's Thesis	13b. Time covered From To	14. Date of Report (*year, month, day*) March 1990	15. Page Count 123

16. Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. Cosati Codes

Field	Group	Subgroup

18. Subject Terms (*continue on reverse if necessary and identify by block number*)

APL2 Editor, Spreadsheet, Frequency Tables, Crosstabulation, Draftsman's Display

19. Abstract (*continue on reverse if necessary and identify by block number*)

A full-screen, scrollable spreadsheet-like editor written in the APL2 language is described for inputting, examining, and outputting data. Mixed numeric and character arrays can be read into or read out to formatted or comma delimited ASCII files. Alternatively a bulk mode input facility allows for rapid direct data entry, or data can be examined and edited cell-by-cell in the usual way. Columns, rows or blocks of data can be highlighted in a chosen color, shadowed, moved or copied. In addition APL functions entered on a command line can use the blocks as input or output. A facility for coding missing values is also provided. Output is obtained as a new spreadsheet, or equivalently as an APL2 matrix. In particular two-way crosstabulations of multiple columns are laid out in the spreadsheet like draftsmen's plots to facilitate investigation and explanation of multivariate categorical data. No numerical coding of the data is needed. Flexible printing of arrays is provided, as well as lexicographic sorting of rows.

A specific application of the techniques and the APL2 program is made to a database constructed with the author's assistance and maintained by the Defense Personnel Security Research and Education Center (PERSEREC), Monterey, California. This database is the basis of a large scale study of the Special Background Investigation. The study is designed to evaluate the productivity of investigative sources in developing the necessary information to determine eligibility for access to Sensitive Compartment Information.

20. Distribution/Availability of Abstract ☒ unclassified/unlimited □ same as report □ DTIC users	21. Abstract Security Classification Unclassified	
22a. Name of Responsible Individual Peter A. W. Lewis	22b. Telephone (*include Area code*) (408) 646-2283	22c. Office Symbol 55Lw

Previous editions are obsolete. Security Classification of this Page

Unclassified

Approved for public release; distribution is unlimited.

Draftsmans Displays for Contingency Tables Using
a Full-screen, Scrollable APL2 Spreadsheet Input/Output Editor
with Application to the
PERSEREC Database of Special Background Investigation

by

Uwe H. Steinfeld
Captain, Federal Republic of Germany Army
Diplom-Mathematiker, University of Bielefeld, FRG, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1990

Author: ______________________________
Uwe H. Steinfeld

Approved by: P. Lew______________________________
Peter A. W. Lewis, Thesis Advisor

Laura Johnson______________________________
Laura D. Johnson, Second Reader

P Purdue______________________________
Peter P. Purdue, Chairman
Department of Operations Research

ABSTRACT

A full-screen, scrollable spreadsheet-like editor written in the APL2 language is described for inputting, examining, and outputting data. Mixed numeric and character arrays can be read into or read out to formatted or comma delimited ASCII files. Alternatively a bulk mode input facility allows for rapid direct data entry, or data can be examined and edited cell-by-cell in the usual way. Columns, rows or blocks of data can be highlighted in a chosen color, shadowed, moved or copied. In addition APL functions entered on a command line can use the blocks as input or output. A facility for coding missing values is also provided. Output is obtained as a new spreadsheet, or equivalently as an APL2 matrix. In particular two-way cross-tabulations of multiple columns are laid out in the spreadsheet like draftsmen's plots to facilitate investigation and explanation of multivariate categorical data. No numerical coding of the data is needed. Flexible printing of arrays is provided, as well as lexicographic sorting of rows.

A specific application of the techniques and the APL2 program is made to a database constructed with the author's assistance and maintained by the Defense Personnel Security Research and Education Center (PERSEREC), Monterey, California. This database is the basis of a large scale study of the Special Background Investigation. The study is designed to evaluate the productivity of investigative sources in developing the necessary information to determine eligibility for access to Sensitive Compartment Information.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to assure that the programs are free of computational and logical errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

ACKNOWLEDGMENT

I would like to thank Dr. Al Blum of the IBM Research Labs for his help with questions on the APL2/32 interpreter.

I. INTRODUCTION

A. GENERAL OUTLINE

This thesis has two aspects. The first aspect is the writing of a full-screen, spreadsheet-like data editor written in the APL2 language to perform preliminary numerical analysis for categorical data. APL2 is the IBM Corporation's implementation of some modern APL concepts. The second aspect of the thesis is the application of this editor to a large database consisting mostly of character data.

Categorical (or categorized) data are data which are presented in the form of attributes falling into certain categories or classes. A categorized variable may simply be a convenient classification of a measurable variable into groups. On the other hand, it may not be expressible in terms of an underlying measurable variable at all. For example, we may classify people by (a) their sex, (b) their hair color, (c) their height; (c) is a categorization of a measurable variable, but (a) and (b) are not. Also the hair color may be expressed on an ordered scale from light to dark; this is not so for (a). (a) is referred to as an *unordered* classification and (b) as an *ordered* one [Ref. 1: p. 536]. An extreme case of an unordered classification would be simply a labelling of different samples.

To assist the statistician in examining a set of categorical data, an APL2 workspace, UEDIT, has been written to permit the analysis in a flexible and consistent way of all types of classification by means of recoding, sorting, frequency counting, crosstabulation and contingency table analysis. The use of this workspace and the computer system requirements are described in Appendix C.

The second aspect of this thesis is the specific application of the UEDIT workspace to a database consisting of character and numerical data. This database was constructed, with the author's assistance, and is maintained by Dr. Ralph Carney at the Defense Personnel Security Research and Education Center (PERSEREC) in Monterey, California. It is the basis of a large scale study designed to evaluate the productivity of investigation sources in developing the necessary information needed to determine eligibility for access to Sensitive Compartmented Information (SCI). At this time the database consists of 1173 issue cases where each case contains 64 fields of information; the total number of cases is projected to be 15,000.

B. PURPOSE OF THE THESIS

The purpose of this thesis is twofold: first, to make a contribution to increased productivity during personnel security investigations, and second, to make a contribution to analytical and computational methodology by bringing together in one package various statistical and computational techniques not available elsewhere.

1. Contribution to Increased Productivity

Special Background Investigation (SBI) is a main component in the process of determining the eligibility for access to Sensitive Compartmented Information. This investigation has to cover a long time in the life of the individual and is expensive. The DCI Personnel Security Working Group (PSWG) is examining the investigative requirements in a large scale study to make a 99% risk assessment on the length of coverage required and to evaluate the productivity, i.e., the importance and usefulness, of the information provided. The study will involve 15,000 issue cases in its final stage.

2. Contribution to Analytical Methodology

There is a great number of statistical software on the market today. The most popular programs, including *STATGRAPHICS, SAS, SPSS/PC+* and *MINITAB*, are all adept at numerical methods. However no one program has a convenient method for tabulating frequency counts, crosstabulation, aggregation of classes of frequencies and recoding of character-based categorical data. The UEDIT APL2 workspace written for this thesis has these capabilities and allows the user to work in a consistent way with one spreadsheet-like matrix containing all the data.

Several functions of the UEDIT workspace are not found in other statistical packages. These functions include the ability to enter new data or change existing data manually in a bulk mode or to import data from formatted or comma delimited ASCII files or, with a separate utility function, data files created by *STATGRAPHICS*. Columns of data can be easily recoded to provide more meaningful labels. Major-to-minor (lexicographic) sorts can be performed on selected columns, conditional and unconditional frequency tabulations and crosstabulations can be performed. During these tasks classes of a categorical variable can be aggregated (pooled) interactively. This is an important part of contingency table analysis, and no other statistical package provides the facility for doing this which is found in UEDIT.[1] The contingency table analysis is performed automatically after each aggregation step. All output is obtained as a new spreadsheet, or equivalently as an APL2 matrix, which is overlaid over the original data and can be edited in the usual way.

C. COMPUTATIONAL TOOLS

The APL2 programming language was chosen for the software development because of its compact code and its ability to handle mixed data arrays, i.e., data in vector or matrix form where each element may be numeric or of character type. Subroutines that take many lines of code in languages like FORTRAN can usually be accomplished with a single line of APL2 code. It is also much easier, due to the organization of an APL2 workspace, to structure large programs by performing certain tasks within subfunctions which can be tested and debugged alone, without the need to recompile the complete program.

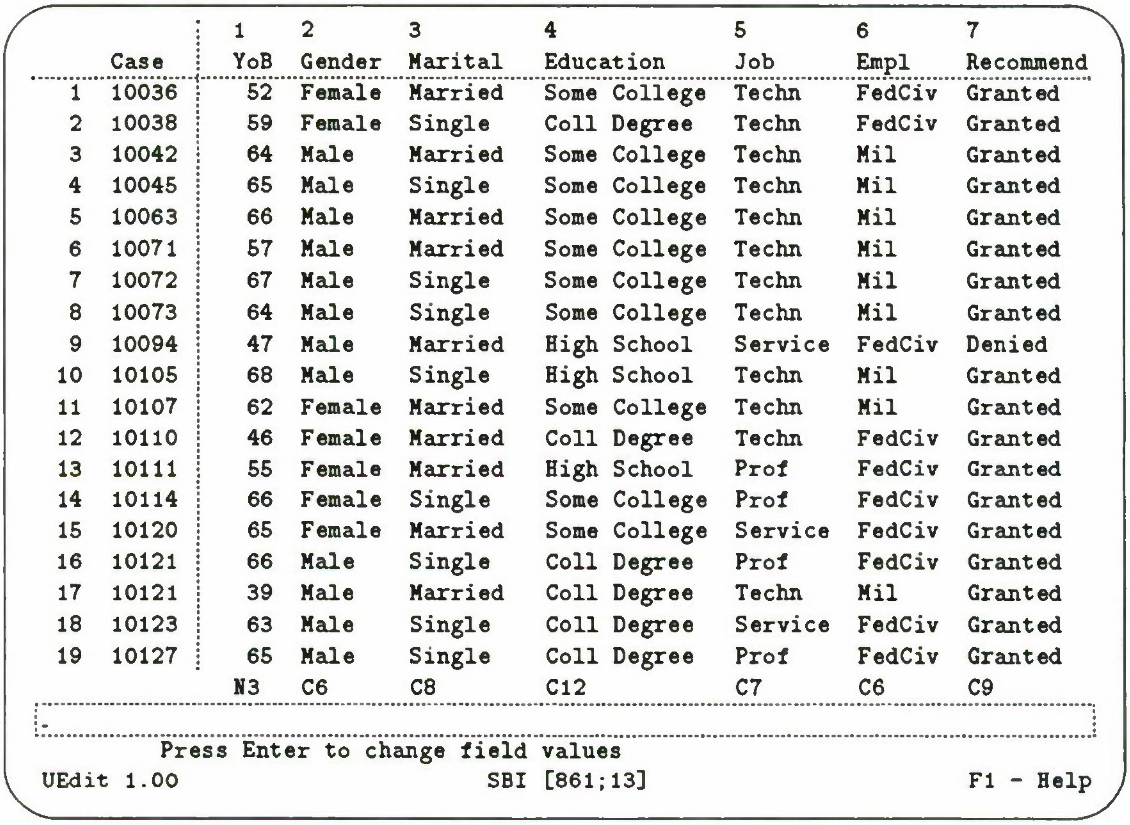
The APL2 interpreter used for this thesis is APL2/32 for the IBM PC (version 1.02) developed by IBM. This interpreter requires a personal computer based on the 80386 microprocessor and the 80387 mathematical coprocessor. Its primary advantage is the ability to use all available random access memory (RAM) without the 640 KB limitation imposed by DOS[2]. However, UEDIT will also work under IBM's APL2/PC which runs on 8086 and 80286 computers. The package uses several *Auxiliary Processors* for file and full-screen display management and for printing. The use of STSC's APL*PLUS II was considered but was rejected because of APL2's more economic memory management, language combatibility to the mainframe version of APL2 and the availabilty of a graphics package, *GRAFSTAT*, for microcomputers in the near future.

[1] For instance, the popular program *STATGRAPHICS* does not provide for aggregation in its contingency table routines. This gap was filled by Ian H. Keith with his APL*PLUS workspace ANALYZE. [Ref. 2]

[2] The term "DOS" is used throughout this thesis as a synonym for both the *Microsoft Disk Operating System* (MS-DOS®) and the *IBM Personal Computer Disk Operating System* (PC-DOS®).

The UEDIT workspace does not require any additional software. However, the PC-version of *GRAFSTAT* which will be released in the near future will provide useful functions, especially for graphical analysis. A fundamental function to export data to *GRAFSTAT* is implemented into UEDIT. In the meanwhile, a statistical package written by Mr. Norman Thomson of the IBM Winchester Laboratories provides a flexible source of routines for use with UEDIT.

In a similar way an interface is provided to *StatXact*, a statistical software package for exact nonparametric inference from Cytel Software Corporation. This interface writes data columns to disk in a format which can be read by *StatXact*.



```
                    :  1   2       3        4              5        6        7
         Case       :  YoB Gender  Marital  Education      Job      Empl     Recommend
....................:...................................................................
    1    10036      :   52 Female  Married  Some College   Techn    FedCiv   Granted
    2    10038      :   59 Female  Single   Coll Degree    Techn    FedCiv   Granted
    3    10042      :   64 Male    Married  Some College   Techn    Mil      Granted
    4    10045      :   65 Male    Single   Some College   Techn    Mil      Granted
    5    10063      :   66 Male    Married  Some College   Techn    Mil      Granted
    6    10071      :   57 Male    Married  Some College   Techn    Mil      Granted
    7    10072      :   67 Male    Single   Some College   Techn    Mil      Granted
    8    10073      :   64 Male    Single   Some College   Techn    Mil      Granted
    9    10094      :   47 Male    Married  High School    Service  FedCiv   Denied
   10    10105      :   68 Male    Single   High School    Techn    Mil      Granted
   11    10107      :   62 Female  Married  Some College   Techn    Mil      Granted
   12    10110      :   46 Female  Married  Coll Degree    Techn    FedCiv   Granted
   13    10111      :   55 Female  Married  High School    Prof     FedCiv   Granted
   14    10114      :   66 Female  Single   Some College   Prof     FedCiv   Granted
   15    10120      :   65 Female  Married  Some College   Service  FedCiv   Granted
   16    10121      :   66 Male    Single   Coll Degree    Prof     FedCiv   Granted
   17    10121      :   39 Male    Married  Coll Degree    Techn    Mil      Granted
   18    10123      :   63 Male    Single   Coll Degree    Service  FedCiv   Granted
   19    10127      :   65 Male    Single   Coll Degree    Prof     FedCiv   Granted
                       N3  C6      C8       C12            C7       C6       C9
:..........................................................................................:
:-                                                                                         :
:..........................................................................................:
          Press Enter to change field values
 UEdit 1.00                        SBI [861;13]                                 F1 - Help
```

Figure 1. Sample UEDIT Full–Screen Data Editor Display: It shows a part of the SBI database with row and column labels. The bottom four lines show the status line with name and size of the edited matrix, the message line where UEDIT displays its prompts and error messages, the user input line and the types and widths of the matrix columns.

II. ANALYSIS OF CATEGORICAL DATA

A. CODING TECHNIQUES

Categorized data may be represented in character or numerical form. Some statistical software packages require that the data to be analyzed have numerical form. Therefore certain attributes in a database have to be recoded by a numeral. For example, in a data column denoting the marital status of an individual, *single* may be coded as "1", *married* as "2", *divorced* as "3" and so on. The absence of an attribute ("unknown" value) could be coded as "0". Normally a character representation is preferred because very often the context cannot be inferred from the numerical representation alone without the help of an additional list explaining the various codes.

UEDIT does not require any numerical coding because each column of a database can contain character data of arbitrary width, or numerical data in fixed decimal, floating point or date representation. For example, the marital status in the SBI database is denoted by "Single", "Married", "Divorced", "Separated", "Widowed", or a blank field if the status is not known. This approach makes it easy to make accurate inferences and to avoid the confusion which might occur when handling large databases containing only numeric codes.

B. FREQUENCY COUNTS

Frequency counting is the determination of the unique elements of an attribute and the number of occurrences of each of those elements. The preferred method for computing the frequency counts is to sort the elements of a sample of size n numerically or lexicographically. Then the first element in the ordered list is used as a label and the list is searched until an element k is found which is not equal to element 1. Now there are $k - 1$ occurrences of element 1 and element k becomes the next unique element. This procedure is repeated until the list is exhausted. The result is a vector $\{f_i,\ i = 1, \ldots, r\}$ of frequencies of r unique elements in the sample. A simple implementation in APL2 may look like

```
I←(U←U[sort ((ρU),1)ρU←((A⍳A)=⍳ρA)/A])⍳A
F←(ρU)ρ0
```

```
F[A/I]←B-(ρB)ρ0,B←(A←(¯1↓I≠1⌽I),1)/⍳ρI←I[⍋I]
```

which takes a vector A of character elements and creates a new vector U which consists of the unique elements of A and a numerical vector F which contains the number of occurrences of each element of U. In line 1 `sort` is a subfunction which sorts the elements of U in any desired order (ascending or descending, case sensitive or case insensitive).

From these (absolute) frequencies f_i relative frequencies $r_i = f_i/n$ and cumulative frequencies $c_i = \sum_{j=1}^{i} f_j$ can easily be calculated by

```
R←F/ρF
C←+\F
```

All theses values are automatically calculated by UEDIT whenever a frequency count is performed.

C. CROSSTABULATION

Crosstabulation is an extension of frequency counting to two-dimensional samples. Two categorized variables with equal number n of elements in a one-to-one correspondence are *crosstabulated* in the following way. Assuming they have r and c unique elements, respectively, then for each unique element i of the first variable a separate frequency count for the number of occurrences of each unique element j of the second variable is performed. This results in a matrix of observed frequencies

$$\begin{array}{ccccc}
n_{11} & n_{12} & \cdots & n_{1c} & n_{1\cdot} \\
n_{21} & n_{22} & \cdots & n_{2c} & n_{2\cdot} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
n_{r1} & n_{22} & \cdots & n_{rc} & n_{r\cdot} \\
n_{\cdot 1} & n_{\cdot 2} & \cdots & n_{\cdot c} & n
\end{array}$$

which is also known as a *contingency table.*

The $n_{i\cdot} = \sum_{j=1}^{c} n_{ij}$ are called row sums or row marginals, the $n_{\cdot j} = \sum_{i=1}^{r} n_{ij}$ are called column sums or column marginals. They represent the frequency counts of both elements by its own, i.e., marginal totals. Again, n is the sample size.

A very efficient method to implement a crosstabulation into a computer program is the *two-way plus reduction* [Ref. 3: p. 97]. It replaces the indices $i = 1, \ldots, r$ of the unique elements of the first variable by $i' = c(i-1)$ where c again is the number of unique elements of the second variable. Then $k = i' + j$ is unique for all $i = 1, \ldots, r$ and $j = 1, \ldots, c$, and a vector $\{f'_k,\ k = 1, \ldots, rc\}$ can be constructed using a frequency tabulation as described before, which consists of the combined frequencies of both variables. By reshaping this vector into a $r \times c$ matrix a contingency table is obtained to which a row of column marginals and a column of row marginals can be added.

D. CONTINGENCY TABLE ANALYSIS

Contingency table analysis is a statistical method to measure and test the interdependence of two categorized variables, or, as it is generally known, the problem of *association*. [Ref. 1: p. 536]

When observations are based on a nominal measurement scale, i.e., have no natural numeric value, distributional measures such as mean or variance are undefined. Association, i.e., dependence, of categorized variables cannot be measured by moments based on a joint probability distribution [Ref. 4: p. 14]. The method of contingency table analysis is very useful for the analysis of associations of categorized variables. It uses the contingency table created by a crosstabulation of two elements.

1. The Chi-Square Test for Independence

Let the observations of a random sample of size n be classified according to two criteria, so that each observation is associated with one of r classes of criterion 1 and one of c classes of criterion 2. Let n_{ij} be the number of observations associated with class i of criterion 1 and class j of criterion 2. Then the n_{ij} can be arranged in an $r \times c$ contingency table as defined in section C.

The assumptions being made are

1. The sample of n observations is a random sample, i.e., each observation has the same probability as every other observation of being classified in row i and column j, independently of the other observations.
2. Each observation may be classified into exactly one of r different categories according to one criterion and into exactly one of c different categories according to the second criterion. [Ref. 5: p. 155]

Then the null hypothesis can be stated as

H_0 : The event "an observation is in row i" is independent of the event "that same observation is in column j", for all i and j.

against the alternative

H_1 : There is lack of independence, i.e., an association exists between at least one column and one row.

By the definition of independence of events, the hypotheses may be stated as follows

$$H_0 : \quad p_{ij} = p_{i\cdot}p_{\cdot j} \qquad \text{for all } i,j$$
$$H_1 : \quad p_{ij} \neq p_{i\cdot}p_{\cdot j} \qquad \text{for some } i,j$$

where p_{ij} is the probability that an observation is in row i and column j, and $p_{i\cdot}$ and $p_{\cdot j}$ are the marginal probabilities of observations in row i and column j respectively.

The most common test statistic for this hypothesis is given by

$$X^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(n_{ij}-e_{ij})^2}{e_{ij}}$$

where n_{ij} are the *observed* frequencies in row i and column j and the e_{ij} denote the *expected* frequencies in row i and column j defined by

$$e_{ij} = \frac{n_{i\cdot}n_{\cdot j}}{n}$$

Here $n_{i\cdot}$ and $n_{\cdot j}$ are the row and column marginals, respectively, and n the total number of observations.

As $n \to \infty$, X^2 is asymptotically distributed as a χ^2 random variable with $(r-1)(c-1)$ degrees of freedom [Ref. 5: p. 156]. Hence the decision rule is to reject H_0 if X^2 exceeds the $1-\alpha$ quantile of a χ^2 random variable with $(r-1)(c-1)$ degrees of freedom where "α is the *maximum* probability of rejecting H_0 when H_0 is true" [Ref. 5: p. 79]. Equivalently the minimum probability of accepting H_0, making the correct decision, is $1-\alpha$. The quantity α is usually called the *level of significance.*

The value $\alpha = \hat{\alpha}$ such that X^2 *equals* the $1-\hat{\alpha}$ quantile of a χ^2 distribution is called the *critical value* or p-value. That is, "$\hat{\alpha}$ is the smallest significance level at which the null hypothesis would be rejected for a given observation". [Ref. 5: p. 81]

However it is important to keep in mind that X^2 is only a test value for the null hypothesis and not a measure of the degree of dependence or independence. Greater values of X^2 do not necessarily imply greater dependence. For a discussion of an adequate measurement of the degree of dependence see Goodman and Kruskal [Ref. 6].

2. Residual Analysis

If the null hypothesis is rejected, i.e., if the test statistic X^2 exceeds the $1 - \alpha$ quantile of the corresponding χ^2 distribution the question arises which combinations of the two variable categories contribute to this decision. The ***standardized residuals*** of observed frequencies, defined by

$$z_{ij} = \frac{n_{ij} - e_{ij}}{\sqrt{e_{ij}}}$$

are useful in this regard as "large" values are an indicator of the degree of contribution. Note that X^2 is equal to the sum over all i and j of z_{ij}^2. [Ref. 7: p. 348] UEDIT uses the built-in shadowing capability to highlight two categories of extremely large residuals to help in their visual recognition.

E. AGGREGATION OF CATEGORIES

An area of ongoing research is the minimum expected frequency that should occur in a contingency table for the asymptotic distribution theory of the χ^2 test statistic to be valid. If the e_{ij} are "too small" then the test statistic X^2 is not well approximated by a χ^2 distribution. Cells with small expectation "are less informative than cells with expectation greater than, say, 3". If $e_{ij} < 2$ "the lowest count of 0 is not improbably low, and the only way for the count to differ 'significantly' from e_{ij} is by being too high". [Ref. 7: p. 291]

One common rule is that no cell should have an expected frequency less than 1 and no more than 1/5 of the cells should have an expectation less than 5. If χ^2 has only 1 degree of freedom, i.e., in a 2×2 table, no cell should have an expectation less than 5 and the total number of observations should be at least 30. [Ref. 8: p. 334]

Another rule uses the *average* expected frequencies rather than the minimum expectations. The average expected frequency $\bar{E}$, is defined as the ratio of the sample size to the number of cells, i.e.,

$$\bar{E} = \frac{n}{rc}$$

The recommendations are

1. If $\alpha = .05$ and cells are equiprobable, then $\bar{E} \geq 1$.
2. If $\alpha = .01$ and cells are equiprobable, then $\bar{E} \geq 2$.
3. If the cells are *not* equiprobable then average expectations should be doubled, i.e., $\bar{E} \geq 2$ for $\alpha = .05$, and $\bar{E} \geq 4$ for $\alpha = .01$. [Ref. 9: p. 758]

If these conditions are not met groups with low expectations should be *aggregated* (pooled) to raise the accuracy of the analysis if an aggregation is feasible. That is, rows or columns with similar context are merged to increase the cell expectation. In most cases aggregation will affect the critical level $\hat{\alpha}$ and may even change the decision whether to reject or accept the null hypothesis. UEDIT leaves the judgment regarding the feasibility of an aggregation and the validity of the test to the user, but allows an aggregation of any classes of a variable and then performs the necessary recalculation automatically.[3]

F. DRAFTSMAN'S TABLES

Contingency tables as described before are also called *two-way* contingency tables where the term "two-way" refers to the number of dimensions of the table or equivalently to the number of variables crosstabulated. An extension may be made to increase the number of variables to three or more which leads to an n-way contingency table with n dimensions [Ref. 5: p. 165]. For example, in the SBI database one might be interested whether the variables GENDER, MARITAL and EDUCATION are independent. This would result in a contingency table with $r = 2$ rows, $c = 5$ columns and $b = 5$ blocks and a test statistic X^2 with a summation over all $r \times c \times b$ cells. X^2 is then tested for significance using the χ^2 distribution with $(r-1)(c-1)(b-1)$ degrees of freedom.

However, with the number of dimensions the difficulty in visualizing the associations and interactions between variables will also increase. Therefore this method is not used very often and is currently not implemented in UEDIT.

A more useful approach is to crosstabulate the n variables of interest pairwise and arrange the $n(n-1)/2$ contingency tables in a *draftsman's table* [Ref. 10: p. 136], an array

[3] Exact tests are available in the *StatXact* package, but its import capability is limited to casefiles with a sample size of maximal 200 and to 2×2 contingency tables.

of tables arranged in the following order

$$\begin{array}{ccccc}
V_1 \text{ vs. } V_2 & V_1 \text{ vs. } V_3 & V_1 \text{ vs. } V_4 & \ldots & V_1 \text{ vs. } V_n \\
 & V_2 \text{ vs. } V_3 & V_2 \text{ vs. } V_4 & \ldots & V_2 \text{ vs. } V_n \\
 & & V_3 \text{ vs. } V_4 & \ldots & V_3 \text{ vs. } V_n \\
 & & & \ddots & \vdots \\
 & & & & V_{n-1} \text{ vs. } V_n
\end{array}$$

where V_i vs. V_j denotes the result of a crosstabulation of variables i and j. Examples are given in Appendix D.

III. CONSTRUCTION OF THE SBI DATABASE

A. BACKGROUND

The Director of Central Intelligence (DCI) Personnel Security Working Group (PSWG) is examining the investigative requirements of the DCI Directive 1/14 with a large scale study of the Special Background Investigation (SBI). The study is designed to evaluate the productivity of investigative sources in developing the necessary information to determine eligibility for access to Sensitive Compartment Information (SCI). The objectives of the study are to:

1. Determine the productivity of sources of information in personnel security investigations.
2. Evaluate the length of coverage needed to determine with reasonable probability that an indication of significant adverse information will be developed.

It is recognized that when significant information is revealed, an inquiry is normally expanded to completely resolve an issue. The purpose of the present PSWG study is to determine which sources provide the first indication of a problem and the minimum period of coverage needed to reveal a problem.

B. DATABASE FORMAT

The data for the PSWG study is recorded by adjudicators on machine-scannable, case summary forms after an initial determination has been reached. These forms are then scanned, and the information is converted into *SAS* format and recorded on tape. To edit and work interactively with the data the tape contents are then dumped to a CMS file on the mainframe computer of the Naval Postgraduate School.

The original database presently consists of 861 records, one for each issue case. The data were collected during the first six months of 1989. Each record contains 166 fields of information pertaining to the particular issue case. The first 13 fields contain the case number and data common to each of up to three issues which are coded in the remaining 153 record fields. The record format is fixed but with variable field lengths.

This data file was then transferred to a PC and preprocessed with *KEDIT* and *Personal REXX*[4] with the purpose of

1. Constructing new records consisting of the 13 common fields and one issue (51 fields) per record. The new database has now 1173 records for 861 individuals, with 64 fields each.
2. Reducing the amount of disk storage needed by converting the file format to "comma delimited", i.e., fields are no longer padded with blanks to preserve the fixed format but are delimited with commas only.

The file was then loaded into UEDIT and recoded using the recoding facility ⟨Shift-F8⟩ provided by the program. Because of the structure of the summary forms and the automatic scanning process all fields in the original database are numeric. In addition some of the fields were coded in binary form, i.e., in a sequence of ones (marked) and zeroes (unmarked), whereas most other fields were coded in decimal form, e.g., the field YEAR OF BIRTH. All categorical fields were recoded to obtain their original label, e.g., in the EDUCATION column

1 → Non-HS
2 → GED
3 → High school
4 → Some college
5 → College degree
6 → Post-graduate

Missing data (unknown values) were consistently coded as a blank field which is internally stored by UEDIT as −32768 for numeric columns as a default value.

To facilitate the analysis, two databases were generated consisting of the *common data* (861 records) and the *issue data* (1173 records), respectively, with the case numbers as identifier fields in both files.

The common data consist mostly of identification fields and demographic attributes. A preliminary analysis of these demographic variables is the objective of the following chapter.

[4] *KEDIT* is a full-screen text editor, *Personal REXX* a computer language. Both are highly compatible to IBM's *XEDIT* and *REXX* respectively which are available on the Naval Postgraduate School mainframe computer.

IV. ANALYSIS OF THE SBI DATABASE

Six demographic variables are included in the database of Security Background Investigation. These are

1. YEAR OF BIRTH
2. GENDER
3. MARITAL STATUS
4. EDUCATION (highest education level of the subject)
5. JOB CATEGORY which is described by
    (a) Professional — project managers, scientists, analysts, military officers, etc.
    (b) Technical — persons involved in the manufacture, operation or maintenance of equipment and military enlisted personnel
    (c) Clerical — persons involved in clerical duties
    (d) Service — charforce, security guards, etc.
6. TYPE OF EMPLOYEE (military, federal civilian or industrial contractor)

A. FREQUENCY COUNTS

The database of the demographic data presently consists of 861 individuals of whom 2/3 are male, 48% are single and 43% are married. 79% are between 18 and 40 years old. Nearly all have at least a high school degree. The complete results are tabulated in Appendix A.

The number of incomplete data, i.e., data that are coded as unknown, which usually causes problems for an accurate analysis is small and mostly negligible, as Table 1 shows.

TABLE 1. PERCENTAGE OF MISSING DATA

Category	Missing data
GENDER	0.2 %
YEAR OF BIRTH	0.1 %
MARITAL STATUS	1.0 %
EDUCATION	1.7 %
JOB CATEGORY	3.9 %
TYPE OF CATEGORY	0.5 %

B. CONTINGENCY TABLES

The columns of demographic data were pairwise crosstabulated using the "Draftsman's Display" option of UEDIT. The resulting contingency tables are listed in Appendix B. Note that the spreadsheet computer display has several advantages over the printed display: observed frequencies and residuals are displayed in different colors, significant residuals are highlighted which makes comparisons much easier.

All tables have χ^2-statistics between 29.6 and 530.5 with p-values less than 0.0011 contradicting the hypothesis of no association between these attributes. Of course, many of the associations found are obvious. For example, only few individuals of age 20 and less are married or have a Postgraduate degree. Only three subjects are widowed so that no significant statements for them can be made. The observations made are summarized below.

YEAR OF BIRTH vs. GENDER

The proportions of males and females in the database are maintained only in the range of ages 30–40. Below this range there are relatively more females, above this range more males.

YEAR OF BIRTH vs. MARITAL STATUS, EDUCATION and JOB CATEGORY

As mentioned before there is a strong tendency for younger individuals to still be single and have not yet finished their education. Therefore additional crosstabulations were done in which the year of birth was conditioned to those that lie in the range 20–59. These results were quite different and showed no strong associations between the categories.

Year of Birth vs.	χ^2	p-level
Marital Status	51.06	.25
Education	46.18	.77
Job Category	33.09	.61

YEAR OF BIRTH vs. TYPE OF EMPLOYEE

The same restricted additional crosstabulation was performed for YEAR OF BIRTH and TYPE OF EMPLOYEE. This limited the amount of associations. The results were $\chi^2 = 31.04$ and a p-level of .03. But a relatively large number of contractors in the range of age 55–65 remains noticeable.

MARITAL STATUS vs. GENDER

The number of married females in the database is relatively low, while there are more married men than the expectation under the assumption of independence.

MARITAL STATUS vs. JOB CATEGORY

This table mirrors the results of the crosstabulations of YEAR OF BIRTH vs. MARITAL STATUS and JOB CATEGORY. For example, singles are usually younger than married individuals and are more likely to fall into the job category Technical.

MARITAL STATUS vs. TYPE OF EMPLOYEE

The most noticable positive relationship in the table is between contractors and married subjects, while there are relatively more single federal civilians and military personnel. Also the rate of divorced contractors and federal civilians is higher than the rate of military personnel.

EDUCATION vs. GENDER

Generally the education level of females tends to be lower than of males.

EDUCATION vs. MARITAL STATUS

Weak associations can be observed between many singles with high school degree and married subjects with postgraduate degree on one hand and few singles with postgraduate degree and married inviduals with high school degree on the other hand.

EDUCATION vs. JOB CATEGORY

As expected there are strong association between these two categories depending on the prerequisites for the different job categories. This resulted in a χ^2-statistic of 442.5.

EDUCATION vs. TYPE OF EMPLOYEE

Generally the education level of contractors and federal civilians in the database is higher than that of military personnel. This may be because the age is correlated with both categories.

JOB CATEGORY vs. GENDER

A very large number of females is involved in clerical duties. This is already obvious by looking at the absolute numbers.

JOB CATEGORY vs. TYPE OF EMPLOYEE

Obviously most military personnel are in the job category Technical which involves all enlisted personnel, while federal civilians are mostly in the other three categories. The result was the highest χ^2 statistic of 530.5.

TYPE OF EMPLOYEE vs. GENDER

Most females are in the category Federal Civilian, while males are more associated with the other classes Contractor and Military.

V. CONCLUSIONS AND RECOMMENDATIONS

A. RESULTS OF THE ANALYSIS

The database of Security Background Investigation with its 861 × 64 fields could easily be handled with UEDIT. The ability to investigate character and numeric data in one matrix by scrolling across the worksheet was extremely useful in enabling the investigation to see characteristics of the data at a glance. Inconsistent coding schemes were spotted at once and recoding was a matter of seconds. The fear that the speed of the editing and the statistical functions would be unacceptable due to the interpretative character of the APL language was soon dismissed.

APL2's capability to handle nested vectors and matrices as a unit causes the loss in speed for editing the SBI database compared to a 150 × 10 matrix to be minimal, except for the time it takes to load a database into the system.

The preliminary analysis of the demographic data in the SBI database was eased by the fact that the percentage of missing data is low. This raised the statistical significance of the statements considerably. However, one has always to bear in mind that correlation does not imply causation. This was demonstrated when several strong associations lost their significance when the subjects under investigation were limited to age 30 and older.

B. RECOMMENDATIONS FOR FURTHER STUDIES

The frequency counting, crosstabulation and contingency table analysis should be repeated when the SBI database is complete. The methodology will remain appropriate. Because of the interrelatedness among all variables the analysis should involve n-way comparison. When *GRAFSTAT/PC* is released by IBM these investigations should be extended by graphical analysis to further improve the visual effect of the results.

The extra facilities offered by a second generation APL interpreter like APL2 make the APL language even more attractive for statistical analysis. The APL2 environment is ideal for collecting a flexible set of investigative tools. Therefore UEDIT has been designed to be open-ended. New functions can be added easily either by implementing them as an integral part of the system—a template has been provided to achieve this task—or by copying them into the workspace in addition to UEDIT and activate them from UEDIT's command line. Possible enhancements in connection with this thesis may be:

1. Extending the analysis to n-way contingency tables to examine multi-way interactions between three or more variables.
2. Providing methods to measure the degree of dependence or independence in contingency tables.
3. Graphical methods for the analysis of contingency tables. This could be done by enhancing the interface to *GRAFSTAT/PC* after its release. The interface is presently kept very rudimentary as no exact specifications are yet available.

APPENDIX A. SBI DATABASE FREQUENCY TABLES

The following tables show the frequency tables for the demographic variables in the SBI database. Presently there are 861 cases in the database. Therefore these tabulations will change after all data have been collected. Unknown attributes are indicated by a dash.

The bar charts in the last table columns are presented in the way produced by UEDIT: If all absolute frequencies are smaller than 40 then all bar lengths (measured in display columns) are equal to those frequencies. Otherwise the longest bar is 40 columns long and the others have lengths proportional to it. The lengths are at least 1 unless there are no observations in a class.

YEAR OF BIRTH

	Freq.	Rel.	Cum.	
–	1	.00	1	□
20–29	7	.01	8	□□
30–34	26	.03	34	□□□□□
35–39	19	.02	53	□□□□
40–44	46	.05	99	□□□□□□□□
45–49	60	.07	159	□□□□□□□□□□□
50–54	85	.10	244	□□□□□□□□□□□□□□□
55–59	114	.13	358	□□□□□□□□□□□□□□□□□□□□
60–64	189	.22	547	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
65–69	236	.27	783	□□
70–72	78	.09	861	□□□□□□□□□□□□□□

GENDER

	Freq	Rel.	Cum.	
–	2	.00	2	□
Female	298	.35	300	□□□□□□□□□□□□□□□□□□□□□□
Male	561	.65	861	□□

MARITAL STATUS

	Freq.	Rel.	Cum.	
–	9	.01	9	□
Single	412	.48	421	□□
Married	374	.43	795	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Separated	12	.01	807	□□
Divorced	51	.06	858	□□□□□
Widowed	3	.00	861	□

EDUCATION

	Freq.	Rel.	Cum.	
–	15	.02	15	□□□
Non HS	8	.01	23	□□
GED	8	.01	31	□□
Highschool	261	.30	292	□□
Some college	229	.27	521	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
College degree	239	.28	760	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Post-graduate	101	.12	861	□□□□□□□□□□□□□□□□

JOB CATEGORY

	Freq.	Rel.	Cum.	
–	34	.04	34	□□□□
Clerical	108	.13	142	□□□□□□□□□□□□□
Professional	340	.39	482	□□
Service	69	.08	551	□□□□□□□□□
Technical	310	.36	861	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

TYPE OF EMPLOYEE

	Freq.	Rel.	Cum.	
–	4	.00	4	□
Contractor	168	.20	172	□□□□□□□□□□□□□□□□□□□
Federal Civilian	356	.41	528	□□
Military	333	.39	861	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

APPENDIX B. SBI DATABASE CONTINGENCY TABLES

The following tables show the results of a pairwise crosstabulation of the demographic data in the SBI database. The 15 tables were created using the "Draftsman's Display" option of UEDIT. The first entry in each field shows the obeserved frequency, the second field shows the standard residual for each pair of attributes. The spreadsheet screen display of UEDIT shows observed frequencies and residuals in different colors, significant residuals are highlighted which makes comparisons much easier. Clearly this highlighting cannot be shown on a black and white page. However it is possible to display only the highlighted residuals by pressing ⟨Shift-S⟩ ("Shadow"). An example of this reduction of the YEAR OF BIRTH vs. GENDER crosstabulation is given below.

YEAR OF BIRTH – GENDER

	–	Female	Male	total
–	.00	.00	1.00	1.00
	-.05	-.59	.43	.00
20–29	.00	1.00	6.00	7.00
	-.13	-.91	.67	.01
30–34	.00	10.00	16.00	26.00
	-.25	.33	-.23	.03
35–39	1.00	1.00	17.00	19.00
	4.55	-2.17	1.31	.02
40–44	.00	9.00	37.00	46.00
	-.33	-1.73	1.28	.05
45–49	.00	14.00	46.00	60.00
	-.37	-1.48	1.10	.07
50–54	.00	30.00	55.00	85.00
	-.44	.11	-.05	.10
55–59	1.00	39.00	74.00	114.00
	1.43	-.07	-.03	.13
60–64	.00	76.00	113.00	189.00
	-.66	1.31	-.91	.22
65–69	.00	84.00	152.00	236.00
	-.74	.26	-.14	.27
70–72	.00	34.00	44.00	78.00
	-.43	1.35	-.96	.09
total	2.00	298.00	561.00	861.00
column %	.00	.35	.65	
d.o.f.	20.0000			
Chi-sq	46.3417			
signif	.0007			

YEAR OF BIRTH – GENDER (shadowed)

	–	Female	Male	total
–	-.05	-.59	.43	.00
20–29	-.13	-.91	.67	.01
30–34	-.25	.33	-.23	.03
35–39	4.55	-2.17	1.31	.02
40–44	-.33	-1.73	1.28	.05
45–49	-.37	-1.48	1.10	.07
50–54	-.44	.11	-.05	.10
55–59	1.43	-.07	-.03	.13
60–64	-.66	1.31	-.91	.22
65–69	-.74	.26	-.14	.27
70–72	-.43	1.35	-.96	.09
column %	.00	.35	.65	

YEAR OF BIRTH – MARITAL STATUS

	–	Single	Married	Separated	Divorced	Widowed	total
–	.00	1.00	.00	.00	.00	.00	1.00
	-.10	.75	-.66	-.12	-.24	-.06	.00
20–29	.00	.00	5.00	1.00	1.00	.00	7.00
	-.27	-1.83	1.12	2.89	.91	-.16	.01
30–34	.00	.00	20.00	.00	6.00	.00	26.00
	-.52	-3.53	2.59	-.60	3.59	-.30	.03
35–39	.00	3.00	13.00	.00	2.00	1.00	19.00
	-.45	-2.02	1.65	-.51	.82	3.63	.02
40-44	.00	3.00	37.00	1.00	5.00	.00	46.00
	-.69	-4.05	3.81	.45	1.38	-.40	.05
45–49	2.00	5.00	45.00	2.00	6.00	.00	60.00
	1.73	-4.43	3.71	1.27	1.30	-.46	.07
50–54	1.00	11.00	62.00	2.00	9.00	.00	85.00
	.12	-4.65	4.13	.75	1.77	-.54	.10
55–59	1.00	34.00	70.00	.00	8.00	1.00	114.00
	-.18	-2.78	2.91	-1.26	.48	.96	.13
60–64	.00	91.00	83.00	4.00	11.00	.00	189.00
	-1.41	.06	.10	.84	-.06	-.81	.22
65–69	3.00	190.00	37.00	2.00	3.00	1.00	236.00
	.34	7.25	-6.47	-.71	-2.94	.20	.27
70–72	2.00	74.00	2.00	.00	.00	.00	78.00
	1.31	6.00	-5.48	-1.04	-2.15	-.52	.09
total	9.00	412.00	374.00	12.00	51.00	3.00	861.00
column %	.01	.48	.43	.01	.06	.00	
d.o.f.	50.0000						
Chi-sq	384.9084						
signif	.0000						

YEAR OF BIRTH – EDUCATION

	–	Non HS	GED	Highsch.	SomeColl.	CollDeg.	PostGrad.	total
–	.00 -.13	.00 -.10	.00 -.10	.00 -.55	1.00 1.42	.00 -.53	.00 -.34	1.00 .00
20–29	.00 -.35	.00 -.26	.00 -.26	2.00 -.08	1.00 -.63	1.00 -.68	3.00 2.40	7.00 .01
30–34	2.00 2.30	1.00 1.54	.00 -.49	7.00 -.31	5.00 -.73	6.00 -.45	5.00 1.12	26.00 .03
35–39	2.00 2.90	.00 -.42	.00 -.42	4.00 -.73	4.00 -.47	6.00 .32	3.00 .52	19.00 .02
40–44	.00 -.90	.00 -.65	1.00 .88	10.00 -1.06	11.00 -.35	14.00 .34	10.00 1.98	46.00 .05
45–49	2.00 .93	1.00 .59	.00 -.75	15.00 -.75	12.00 -.99	16.00 -.16	14.00 2.62	60.00 .07
50–54	3.00 1.25	.00 -.89	.00 -.89	16.00 -1.92	25.00 .50	27.00 .70	14.00 1.28	85.00 .10
55–59	3.00 .72	.00 -1.03	3.00 1.89	18.00 -2.82	29.00 -.24	33.00 .24	28.00 4.00	114.00 .13
60–64	2.00 -.71	2.00 .18	2.00 .18	45.00 -1.62	46.00 -.60	68.00 2.15	24.00 .39	189.00 .22
65–69	.00 -2.03	4.00 1.22	.00 -1.48	76.00 .53	88.00 3.18	68.00 .31	.00 -5.26	236.00 .27
70–72	1.00 -.31	.00 -.85	2.00 1.50	68.00 9.12	7.00 -3.02	.00 -4.65	.00 -3.02	78.00 .09
total column %	15.00 .02	8.00 .01	8.00 .01	261.00 .30	229.00 .27	239.00 .28	101.00 .12	861.00
d.o.f. Chi-sq signif	60.0000 265.9455 .0000							

YEAR OF BIRTH – JOB CATEGORY

	–	Clerical	Profess.	Service	Technical	total
–	1.00	.00	.00	.00	.00	1.00
	4.83	-.35	-.63	-.28	-.60	.00
20–29	.00	.00	5.00	.00	2.00	7.00
	-.53	-.94	1.34	-.75	-.33	.01
30–34	2.00	3.00	10.00	4.00	7.00	26.00
	.96	-.14	-.08	1.33	-.77	.03
35–39	2.00	1.00	12.00	1.00	3.00	19.00
	1.44	-.90	1.64	-.42	-1.47	.02
40–44	3.00	8.00	28.00	2.00	5.00	46.00
	.88	.93	2.31	-.88	-2.84	.05
45–49	5.00	5.00	33.00	6.00	11.00	60.00
	1.71	-.92	1.91	.54	-2.28	.07
50–54	2.00	13.00	43.00	6.00	21.00	85.00
	-.74	.72	1.63	-.31	-1.74	.10
55–59	8.00	8.00	60.00	10.00	28.00	114.00
	1.65	-1.67	2.23	.29	-2.04	.13
60–64	7.00	32.00	76.00	17.00	57.00	189.00
	-.17	1.70	.16	.48	-1.34	.22
65–69	3.00	26.00	66.00	16.00	125.00	236.00
	-2.07	-.66	-2.82	-.67	4.34	.27
70–72	1.00	12.00	7.00	7.00	51.00	78.00
	-1.19	.71	-4.29	.30	4.32	.09
total	34.00	108.00	340.00	69.00	310.00	861.00
column %	.04	.13	.39	.08	.36	
d.o.f.	40.0000					
Chi-sq	165.4638					
signif	.0000					

YEAR OF BIRTH – EMPLOYEE TYPE

	–	Contractor	FedCivilian	Military	total
–	.00	1.00	.00	.00	1.00
	-.07	1.82	-.64	-.62	.00
20–29	.00	5.00	2.00	.00	7.00
	-.18	3.11	-.53	-1.65	.01
30–34	.00	17.00	9.00	.00	26.00
	-.35	5.30	-.53	-3.17	.03
35–39	.00	9.00	7.00	3.00	19.00
	-.30	2.75	-.31	-1.60	.02
40–44	.00	19.00	19.00	8.00	46.00
	-.46	3.35	.00	-2.32	.05
45–49	.00	19.00	22.00	19.00	60.00
	-.53	2.13	-.56	-.87	.07
50–54	.00	27.00	34.00	24.00	85.00
	-.63	2.56	-.19	-1.55	.10
55–59	1.00	23.00	48.00	42.00	114.00
	.65	.16	.13	-.31	.13
60–64	3.00	38.00	87.00	61.00	189.00
	2.26	.18	1.00	-1.41	.22
65–69	.00	9.00	109.00	118.00	236.00
	-1.05	-5.46	1.16	2.80	.27
70–72	.00	1.00	19.00	58.00	78.00
	-.60	-3.64	-2.33	5.07	.09
total	4.00	168.00	356.00	333.00	861.00
column %	.00	.20	.41	.39	
d.o.f.	30.0000				
Chi-sq	191.2516				
signif	.0000				

MARITAL STATUS – GENDER

	–	Female	Male	total
–	.00	1.00	8.00	9.00
–	-.14	-1.20	.88	.01
Single	.00	156.00	256.00	412.00
	-.98	1.12	-.76	.48
Married	1.00	107.00	266.00	374.00
	.14	-1.97	1.43	.43
Separated	.00	6.00	6.00	12.00
	-.17	.91	-.65	.01
Divorced	1.00	25.00	25.00	51.00
	2.56	1.75	-1.43	.06
Widowed	.00	3.00	.00	3.00
	-.08	1.93	-1.40	.00
total	2.00	298.00	561.00	861.00
column %	.00	.35	.65	
d.o.f.	10.0000			
Chi-sq	29.5805			
signif	.0010			

MARITAL STATUS – JOB CATEGORY

	–	Clerical	Profess.	Service	Technical	total
–	3.00	.00	1.00	.00	5.00	9.00
	4.44	-1.06	-1.35	-.85	.98	.01
Single	10.00	47.00	152.00	33.00	170.00	412.00
	-1.55	-.65	-.84	.00	1.78	.48
Married	20.00	43.00	170.00	27.00	114.00	374.00
	1.36	-.57	1.84	-.54	-1.78	.43
Separated	.00	2.00	2.00	3.00	5.00	12.00
	-.69	.40	-1.26	2.08	.33	.01
Divorced	1.00	14.00	15.00	6.00	15.00	51.00
	-.71	3.01	-1.15	.95	-.78	.06
Widowed	.00	2.00	.00	.00	1.00	3.00
	-.34	2.65	-1.09	-.49	-.08	.00
total	34.00	108.00	340.00	69.00	310.00	861.00
column %	.04	.13	.39	.08	.36	
d.o.f.	20.0000					
Chi-sq	67.6096					
signif	.0000					

MARITAL STATUS – EMPLOYEE TYPE

	–	Contractor	FedCivilian	Military	total
–	.00	.00	.00	9.00	9.00
	-.20	-1.33	-1.93	2.96	.01
Single	2.00	35.00	197.00	178.00	412.00
	.06	-5.06	2.04	1.48	.48
Married	2.00	117.00	124.00	131.00	374.00
	.20	5.15	-2.46	-1.13	.43
Separated	.00	2.00	6.00	4.00	12.00
	-.24	-.22	.47	-.30	.01
Divorced	.00	14.00	27.00	10.00	51.00
	-.49	1.28	1.29	-2.19	.06
Widowed	.00	.00	2.00	1.00	3.00
	-.12	-.77	.68	-.15	.00
total	4.00	168.00	356.00	333.00	861.00
column %	.00	.20	.41	.39	
d.o.f.	15.0000				
Chi-sq	90.0469				
signif	.0000				

EDUCATION – GENDER

	–	Female	Male	total
–	.00	2.00	13.00	15.00
	-.19	-1.40	1.03	.02
Non HS	.00	4.00	4.00	8.00
	-.14	.74	-.53	.01
GED	.00	3.00	5.00	8.00
	-.14	.14	-.09	.01
Highschool	.00	115.00	146.00	261.00
	-.78	2.60	-1.84	.30
Some College	1.00	85.00	143.00	229.00
	.64	.64	-.51	.27
CollDegree	.00	70.00	169.00	239.00
	-.75	-1.40	1.06	.28
PostGraduate	1.00	19.00	81.00	101.00
	1.58	-2.70	1.87	.12
total	2.00	298.00	561.00	861.00
column %	.00	.35	.65	
d.o.f.	12.0000			
Chi-sq	32.7184			
signif	.0011			

EDUCATION – MARITAL STATUS

	–	Single	Married	Separated	Divorced	Widowed	total
–	4.00 9.71	1.00 -2.31	8.00 .58	.00 -.46	2.00 1.18	.00 -.23	15.00 .02
Non HS	.00 -.29	3.00 -.42	4.00 .28	1.00 2.66	.00 -.69	.00 -.17	8.00 .01
GED	.00 -.29	3.00 -.42	4.00 .28	1.00 2.66	.00 -.69	.00 -.17	8.00 .01
Highschool	2.00 -.44	147.00 1.98	84.00 -2.76	7.00 1.76	18.00 .65	3.00 2.19	261.00 .30
Some College	2.00 -.25	105.00 -.44	108.00 .85	.00 -1.79	14.00 .12	.00 -.89	229.00 .27
CollDegree	1.00 -.95	120.00 .53	106.00 .21	2.00 -.73	10.00 -1.10	.00 -.91	239.00 .28
PostGraduate	.00 -1.03	33.00 -2.21	60.00 2.43	1.00 -.34	7.00 .42	.00 -.59	101.00 .12
total	9.00	412.00	374.00	12.00	51.00	3.00	861.00
column %	.01	.48	.43	.01	.06	.00	
d.o.f.	30.0000						
Chi-sq	158.6933						
signif	.0000						

EDUCATION – JOB CATEGORY

	–	Clerical	Profess.	Service	Technical	total
–	4.00	.00	4.00	.00	7.00	15.00
	4.43	-1.37	-.79	-1.10	.69	.02
Non HS	.00	.00	.00	2.00	6.00	8.00
	-.56	-1.00	-1.78	1.70	1.84	.01
GED	.00	1.00	.00	1.00	6.00	8.00
	-.56	.00	-1.78	.45	1.84	.01
Highschool	8.00	49.00	16.00	33.00	155.00	261.00
	-.72	2.84	-8.58	2.64	6.30	.30
Some College	8.00	44.00	51.00	19.00	107.00	229.00
	-.35	2.85	-4.15	.15	2.70	.27
CollDegree	9.00	12.00	179.00	12.00	27.00	239.00
	-.14	-3.28	8.71	-1.63	-6.37	.28
PostGraduate	5.00	2.00	90.00	2.00	2.00	101.00
	.51	-3.00	7.94	-2.14	-5.70	.12
total	34.00	108.00	340.00	69.00	310.00	861.00
column %	.04	.13	.39	.08	.36	
d.o.f.	24.0000					
Chi-sq	442.2512					
signif	.0000					

EDUCATION - EMPLOYEE TYPE

	–	Contractor	FedCivilian	Military	total
–	.00 -.26	8.00 2.97	1.00 -2.09	6.00 .08	15.00 .02
Non HS	.00 -.19	1.00 -.45	1.00 -1.27	6.00 1.65	8.00 .01
GED	.00 -.19	.00 -1.25	3.00 -.17	5.00 1.08	8.00 .01
Highschool	1.00 -.19	31.00 -2.79	91.00 -1.63	138.00 3.69	261.00 .30
Some College	.00 -1.03	36.00 -1.30	95.00 .03	98.00 1.00	229.00 .27
CollDegree	3.00 1.79	54.00 1.08	115.00 1.63	67.00 -2.65	239.00 .28
PostGraduate	.00 -.69	38.00 4.12	50.00 1.27	13.00 -4.17	101.00 .12
total column %	4.00 .00	168.00 .20	356.00 .41	333.00 .39	861.00
d.o.f. Chi-sq signif	18.0000 98.9490 .0000				

JOB CATEGORY - GENDER

	–	Female	Male	total
–	.00 -.28	8.00 -1.10	26.00 .82	34.00 .04
Clerical	.00 -.50	95.00 9.42	13.00 -6.84	108.00 .13
Professional	2.00 1.36	77.00 -3.75	261.00 2.65	340.00 .39
Service	.00 -.40	22.00 -.39	47.00 .30	69.00 .08
Technical	.00 -.85	96.00 -1.09	214.00 .85	310.00 .36
total column %	2.00 .00	298.00 .35	561.00 .65	861.00
d.o.f. Chi-sq signif	8.0000 163.7666 .0000			

JOB CATEGORY – EMPLOYEE TYPE

	–	Contractor	FedCivilian	Military	total
–	1.00	24.00	4.00	5.00	34.00
	2.12	6.74	-2.68	-2.25	.04
Clerical	.00	4.00	103.00	1.00	108.00
	-.71	-3.72	8.73	-6.31	.13
Professional	2.00	96.00	172.00	70.00	340.00
	.33	3.64	2.65	-5.36	.39
Service	.00	5.00	57.00	7.00	69.00
	-.57	-2.31	5.33	-3.81	.08
Technical	1.00	39.00	20.00	250.00	310.00
	-.37	-2.76	-9.55	11.88	.36
total	4.00	168.00	356.00	333.00	861.00
column %	.00	.20	.41	.39	
d.o.f.	12.0000				
Chi-sq	530.5306				
signif	.0000				

EMPLOYEE TYPE – GENDER

	–	Female	Male	total
–	.00	1.00	3.00	4.00
	-.10	-.33	.24	.00
Contractor	.00	28.00	140.00	168.00
	-.62	-3.95	2.92	.20
FedCivilian	2.00	181.00	173.00	356.00
	1.29	5.21	-3.87	.41
Military	.00	88.00	245.00	333.00
	-.88	-2.54	1.90	.39
total	2.00	298.00	561.00	861.00
column %	.00	.35	.65	
d.o.f.	6.0000			
Chi-sq	79.3027			
signif	.0000			

APPENDIX C. UEDIT USER'S MANUAL

A. SYSTEM REQUIREMENTS

UEDIT is a workspace for IBM's APL2/PC which will run on any IBM-compatible microcomputer with at least 512 KB RAM. However, due to the memory limitations of DOS and the size of UEDIT a database should not contain more than approximately 1000 fields on a machine with full 640 KB RAM. The maximum which can be processed depends on the contents of the fields. Therefore it is recommended for moderately large or large databases that one run the program on an 80386/80387 computer under APL2/32. This APL2 interpreter utilizes all available memory in the machine up to 16 MB.

The program will operate with almost any monochrome or color video adapter with an appropriate monitor. UEDIT supports the EGA 43-line mode and the VGA 50-line mode. To print matrices a printer supported by APL2 is required. A special Auxiliary Processor (AP 81) which enhances the support of printers compatible with the Hewlett Packard LaserJet is provided with UEDIT, together with two soft fonts.

The program was written using version 1.02 of APL2/32 on an 80386 based computer with 4 MB of RAM, an EGA video adapter and monitor and a Hewlett Packard LaserJet II printer.

B. PROGRAM AVAILABILITY

A copy of this workspace is available from the author or Professor Peter A. W. Lewis at the address given in the Initial Distribution List. Please send a 5 1/4 inch or 3 1/2 inch IBM-compatible formatted disk.

C. GETTING STARTED

Before you start APL2 you should create a subdirectory to hold data files created by UEDIT, for example with the DOS command

```
MD C:\APL2\UEDIT
```

This is not absolutely necessary as all files saved by UEDIT can be recognized by their file extension .UED. However, it is recommended that you keep the APL2 program files and the data files in separate directories.

Then start the APL2 interpreter with the Auxiliary Processors AP2 (Non-APL program interface), AP80 or AP81 (Printer control), AP124 (Full-screen display management), AP210 (DOS file management) and AP440 (Sound generator) and load UEDIT. You may use a batch file containing the commands

```
APL2FONT
APL232 AP2 AP81 AP124 AP210 AP440 )LOAD UEDIT
APL2FONT /T
```

where APL2FONT loads/unloads the APL2 video character set.

When you have loaded UEDIT for the first time you should inspect certain global variables which contain default values that may have to be configured for your needs. Do not forget to subsequently save the workspace with the)*SAVE* command to set the new defaults permanently. All of these variables can also be changed temporarily from inside the UEDIT environment. The following subsections describe the variables.

1. PATH

The variable *PATH* contains the default directory path to UEDIT's data directory. Use an assignment like

```
PATH←'C:\APL2\DATA\'
```

to set the path to your needs. Note that the path assignment must be finished with a backslash. All spreadsheets created in UEDIT and saved with ⟨F2⟩ or ⟨Shift-F2⟩ (see page 43) will be saved in this directory with a file extension ".UED".

2. MISS∆̲N

To display a "missing numeric value" (unknown attribute) as a blank field it is necessary to assign a special numeric value to the appropriate field which is unlikely to occur in a database. By default UEDIT uses −32768. To change this value assign a new value to *MISS∆̲N*, e.g.,

```
MISS∆̲N←¯99999
```

This can also be done with UEDIT's function ⟨Ctrl-F9⟩ (see page 48).

3. PRINT

The variable *PRINT* is a 7-element vector containing default parameters for printouts of matrices. See page 54 for a detailed discussion of these values.

4. DATE

Dates are stored as number of days since February 29, 0000, and displayed in the form MM-DD-YYYY by default. The display order can be changed by changing the global variable *DATE* which contains a 3-element vector where YYYY corresponds to 1, MM to 2, DD to 3. Thus the default value of *DATE* is 2 3 1. To change to a European style date display DD-MM-YYY you would assign

```
DATE←3 2 1
```

D. RUNNING UEDIT

To edit an APL2 array *MATRIX* start your UEDIT session with the command

```
UEDIT 'MATRIX'
```

UEDIT then performs the following steps

1. If a file MATRIX.UED exists in the data directory UEDIT reads the matrix and its parameters from this file.
2. Otherwise, if an array *MATRIX* exists in the active workspace UEDIT starts the session with this matrix, creating new format and attribute parameters.
3. Otherwise UEDIT creates a new array *MATRIX* and prompts for a vector of column formats (see below).

You can start a program with

```
UEDIT ''
```

In this case the program goes immediately to the File Operations menu (see page 44) to allow the import of a comma delimited or formatted DOS file.

E. COLUMN FORMATS

The following codes are valid for defining new columns or changing column types.

A	Standard APL numeric format
Nx	Numeric with x decimals
Ex	Scientific format; the mantissa is displayed with x decimals
C	Character format
D	Date format

All column widths are set dynamically depending on the largest field in each column. Note also that the number of decimals is only significant for screen and printer output. Internally all numbers are stored at their full APL2 accuracy.

For example, to create a new matrix which will consist of the columns "Name", "Day of Birth", "Years of Service", "Salaries" you would respond to the prompt for new column formats with

```
C,D,A,N2
```

Note that the elements are separated by commas.

UEDIT always displays the current formats below the matrix columns. An identifier C, N or D representing character, numeric or date data, respectively, is followed by the total column width. If a numeric column has a fixed decimal format, a period with the number of decimals is added. Thus for the example above the display may show

```
C6    D8    N2    N7.2
```

F. MOVING AROUND

When you edit a matrix for the first time the cursor will be located in the first field of the matrix, i.e., in the upper left corner. The cursor position is always one complete field indicated by an inverse video display. When you save your work the cursor position is also saved so that you can resume editing at the position where you stopped.

To move the cursor and the editor window around, several key combinations are available:

1. The cursor keys (←→↑↓) move the cursor one field into the appropriate direction as long as the matrix borders are not yet reached. If necessary the editor window will scroll into the opposite direction to show the new active field.

2. To scroll the matrix by one field within the editor window use ⟨Ctrl-←⟩ and ⟨Ctrl-→⟩ for horizontal moves or ⟨Ctrl-PgUp⟩ and ⟨Ctrl-PgDn⟩ for vertical moves.
3. To scroll the matrix up or down one full window at a time use the ⟨PgUp⟩ and ⟨PgDn⟩ keys. To scroll one window to the left and right use ⟨Tab⟩ and ⟨Shift-Tab⟩.
4. To position the cursor on the matrix edges press ⟨Home⟩ for the first and ⟨End⟩ for the last column, ⟨Ctrl-Home⟩ and ⟨Ctrl-End⟩ for the first and last row, respectively.
5. If you want to locate the cursor in a specific field hit ⟨Ctrl-L⟩. UEDIT will prompt you for the row and column number and position the cursor in that field scrolling the window if necessary.

Reminders of these key combinations are also available on UEDIT's on-line help screens.

G. DATA INPUT AND MODIFICATION OF DATA USING APL2 COMMANDS

The default keyboard layout is "APL mode" which makes several ASCII characters unaccessible in the usual way. To switch the layout to the normal "ASCII mode" (typewriter keyboard) hit ⟨Ctrl-Backspace⟩ or ⟨Alt-Backspace⟩. These key combinations are toggles, i.e., they take you back and forth between the two modes every time you hit them.

When you want to enter a value for the active field, i.e., the field displayed in inverse video, just start typing. Any key which does not invoke a special editing function will be recognized as the first character of a new value for the active field. The "input line", which is the third line from the bottom, is then activated — the color changes to high intensity — and it will accept further input until the ⟨Enter⟩ key is hit. The new value is written into the matrix, the display updated and the input line is closed.

If the active field has a numeric type you can input an expression which has a numeric scalar as its result. Elements of the current matrix can be accessed in this input in several ways: UEDIT works with a copy *MAT* of the original matrix. Thus any element of the current matrix can be used with *MAT*[*i*;*j*] where *i* and *j* are the row and column indices, respectively. A short-cut notation for the element at the cursor position is α. A synonym for a vector of all marked elements of the matrix is ω (see page 41). For example, to double the value of the active cell (in a numeric column, of course) you can type

```
2×α
```

After the ⟨Enter⟩ key is hit the value in the cursor position is doubled. To add the elements of column 1 and assign the sum to the active field you would type

```
+/MAT[;1]
```

followed by ⟨Enter⟩. For additional examples see the section on marking and highlighting of areas (page 41) and the description of function key ⟨Ctrl-F4⟩ (page 45).

To change a field hit ⟨Enter⟩. This will copy the field content to the activated input line and you can edit it by overwriting or inserting characters — use the ⟨Ins⟩ key to toggle between overwrite and insert mode.

Additional keystrokes recognized during the data input are:

⟨Home⟩ which locates the cursor at the beginning of the input line,

⟨End⟩ which locates the cursor at the end of the line and

⟨Escape⟩ which cancels the input, i.e., terminates the input but leaves the field unchanged.

Many functions of UEDIT allow data vectors as input. To separate the elements of a vector you should for consistency always use commas, although very often blanks are also accepted as valid delimiters. If a vector element contains a comma itself enclose the element in double quotes (") if your keyboard layout is set to ASCII mode, or in diereses (¨) if you are working in APL mode.

H. INPUT OF DATES

Dates are internally stored as number of days since February 29, 0000[5]. This allows computations to be performed on a matrix column defined as dates.

Valid date specifications in input mode are (assuming the default order of month-day-year as given by the variable *DATE*)

```
MM-DD-YYYY
MM/DD/YYYY
MM.DD.YYYY
MM DD YYYY
```

[5] This base was chosen because it makes the conversion between internal and display format easy and fast.

You can omit the year. In this case UEDIT will insert the current year which is taken from the DOS system date. If you enter the year with only two digits the current century will be inserted.

Every input is checked for validity. This means, invalid dates like 2–29–1990 or 00–00–1990 will be rejected, and you will be prompted for a correction.

I. MARKING AND HIGHLIGHTING OF MATRIX AREAS

Marking and highlighting of matrix cells are similar actions but with a different philosophy. While marking is used as a preparation for a several editing functions, e.g., to copy, move, print or rotate matrix areas, highlighting is used to emphasize the contents of matrix fields. UEDIT will highlight matrix fields by itself during crosstabulations.

The term "marked area" (or "highlighted area") denotes the smallest submatrix of the original matrix where each row and each column contains at least one marked (highlighted field). That is, it is the original matrix with all rows and columns removed which have no marked or highlighted fields. Note that this area may contain fields which are not marked or highlighted.

1. Marking

To prepare certain fields for editing actions use the following key combinations. The marking is indicated by a different background color and a blinking "M" in the upper left corner of the screen. This is useful as a reminder if the marked fields are scrolled off the screen.

Ctrl-F Mark a single field
Marks the field at the current cursor location. The function works as a toggle, i.e., you can hit ⟨Ctrl-F⟩ again to unmark the field. It also defines the first corner of a marked block (see the next item).

Ctrl-B Mark a block
Defines the second corner of a block to be marked. The first corner was fixed the last time ⟨Ctrl-F⟩ was hit. All fields within the rectangular area defined by the two opposite corners will be marked.

Ctrl-R Mark a row
Marks all fields in the row defined by the current cursor location.

Ctrl-C Mark a column
Marks all fields in the column defined by the current cursor location.

Ctrl-U **Unmark**
Removes all marking information from the matrix.

After you have marked one or more fields of the matrix you can use the symbol ω as a short-cut notation for a vector of these fields. This vector is built in row-major order. That is, when you have marked the fields *MAT*[1;1], *MAT*[1;3], *MAT*[2;1] and *MAT*[2;2], you have implicitly assigned

```
ω←MAT[1;1],MAT[1;3],MAT[2;1],MAT[2;2]
```

For example, if these four fields contain numbers, you can add the first three elements, divide the sum by the last field and assign the result to the active field by typing

```
(+/3↑ω)÷ω[4]
```

as a new data input. More examples can be found in the description of function key ⟨Ctrl-F4⟩ (page 45) and in Appendix D.

2. Highlighting

Six levels of highlighting are available, indicated by different high-intensity foreground colors and a blinking "H" in the upper left corner of the screen. The level of the normal display is 0. You can always change the levels by assigning a new level. The functions are similar to those used for marking and are defined as follows ("Sh" denotes the Shift key).

Sh-0...6 **Highlight a single field**
Highlights a single field in the matrix and assigns a (color) level of 0,...,6 to the field. It also defines the first corner and the color level of a highlighted block. Highlighting to level 0 is the same as removing the highlight information. For consistency with the marking syntax ⟨Sh-F⟩ is available which will prompt you for a color level.

Sh-B **Highlight a block**
Fixes the second corner of a block and highlights this block in the color given by the first corner.

Sh-R **Highlight a row**
Highlights all fields in the row defined by the current cursor location. You will be prompted for the color level.

Sh-C **Highlight a column**
Highlights all fields in the column defined by the current cursor location. You will be prompted for the color level.

Sh-U **Unhighlight**
Removes all highlighting information from the matrix. To unhighlight only certain areas of the matrix use the functions above and assign color level 0.

Sh-S **Shadowing**
This option asks for a color level and then hides all rows and columns which contain only fields below that level. The newly created matrix (including the column and row labels) is overlaid over its "parent" matrix and can be edited in the same way. To return to the original matrix use "Quit" ⟨F3⟩ or "File" ⟨F4⟩ (see page 44). When you use ⟨F4⟩ all changes are entered into the parent matrix.

J. UEDIT FUNCTIONS

Once the session is started the whole range of UEDIT's functions can be accessed using certain keys or key combinations. In the following descriptions a "S-" denotes the ⟨Shift⟩ key, "C-" the ⟨Control⟩ key, i.e., ⟨S-F10⟩ means to hold down the ⟨Shift⟩ key while pressing ⟨F10⟩.

F1 **Help**
This function displays three pages of on-line help. The pages contain short reminders of the definitions of all function keys and the description of the cursor movement keys. Use ⟨PgDn⟩ and ⟨PgUp⟩ to see all pages.

S-F1 **Sort**
Sorts the rows of the matrix simultaneously on any number of columns in major-minor order. The sorting on character columns is performed lexicographically and is case-insensitive, i.e., lower and upper case entries are equivalent. The normal sort order is ascending. To sort descending enter the column numbers with a negative sign. For example, when you enter the column numbers to sort on as

```
1,¯5,10
```

UEDIT first sorts on column 10 in ascending order, then in descending order on column 5, and finally on column 1 (the most significant) in ascending order. Another example is described in Appendix D. When you sort the rows of a frequency table the cumulative frequencies will be updated automatically.

C-F1 **Refresh**
If by some unexpected action the screen becomes fragmented use this function to restore the the correct display of the worksheet.

F2 **Save**
Saves the edited matrix into a file of the same name (limited to the first eight characters) and the file extension .UED and places this file in the data directory as indicated by the global variable *PATH*. A copy of the matrix is kept as a global variable in the active workspace. Matrix attributes (column formats, highlighting and marking information) are also saved in this file. The editing session continues.

S-F2 **Save As...**
Performs the same action as the "Save" function ⟨F2⟩ but prompts for a new matrix/file name. This action allows the user to save a matrix under different names in several stages of the editing process. When the session continues the newly assigned name is the default name.

C-F2 **File Operations**
This option displays a submenu of available functions to import or export files. Note that the export operations do not save matrix attributes, e.g. marked areas. Presently the following file formats are supported:

1. Read/Write formatted ASCII files — All fields in a data column have equal widths padded with blanks if necessary. UEDIT will prompt for the column widths before it reads such a file as there is no way to safely determine them. When it writes a formatted file adjacent columns will be separated by two blanks.
2. Read/Write comma separated files — The fields of a record are delimited by commas. Trailing blanks in a field are not necessary. A field which contains a comma must be enclosed in double quotes ("). This format is supported by most commercial database and spreadsheet programs. It is also the fastest way to import a file into UEDIT. Note that *STATGRAPHICS* can only read comma separated files but not write to them.
3. Export to *GRAFSTAT* — As *GRAFSTAT/PC* has not yet been released by IBM only a basic export capability is provided presently. This will write a matrix column as a variable into the active workspace. The variable will be a vector if the column has numeric format, otherwise it will be a two-dimensional character matrix with one element per row. It can be used as an input to the user's own APL2 functions, unless the user wants to execute them on the command line (see below under ⟨C-F4⟩).
4. Export to *StatXact* — This function writes one or more matrix columns to a DOS file which can be imported into *StatXact*. Line numbers are added automatically by UEDIT. Note that *StatXact* will accept only numeric data and its import capability is limited to samples of up to 200 records and and 2 × 2 contingency tables.

F3 **Quit**
Exits the current editing session without saving the matrix. If the matrix has been changed since the start of the session or the last "Save" ⟨F2⟩ or "Save As" ⟨S-F2⟩ operation you will be prompted to confirm the termination.

S-F3 **Put**
Saves a marked area of the matrix (see page 41) including row and column labels into a new APL2 matrix in the active workspace. This function will not write to disk.

C-F3 **Get**
Inserts or overlays another APL2 matrix from the active workspace into the currently edited matrix. UEDIT will ask whether to insert new rows or columns or to overlay an existing area of the matrix. If you choose to insert rows the shape of the added matrix will be adjusted, i.e., if the new matrix has less columns than the current matrix it will be padded with empty ("missing") columns, if it has more columns the excess columns will be truncated. The corresponding actions are taken when you choose to insert new columns. When you want to overlay the new matrix over the current one the position of the cursor determines the upper left corner of the overlay area. The same adjustments as for insertions are made if necessary.

F4 **File**
Saves the edited matrix into a file of the same name (limited to the first eight characters) and the file extension .UED and places this file in the data directory as indicated by the global variable *PATH* and terminates the editing session. Matrix attributes (column formats, highlighting and marking information) are also saved in this file. A copy of the matrix is held as a global variable in the active workspace.

S-F4 **New Matrix**
Starts a new editing session with a different matrix without leaving the UEDIT environment. This is the same as "Quit" ⟨F3⟩ and then typing *UEDIT 'NEWMATRIX'* in the APL2 environment.

C-F4 **APL command**
This function allows the user to submit any valid APL2 command. It is a simple way to implement additional functions into the UEDIT workspace. Elements of the current matrix can be accessed as described before: any element of the matrix can be used with *MAT*[*i*;*j*] where *i* and *j* are the row and column indices, respectively. A short-cut notation for the element at the cursor position is ⍺. A synonym for a vector of all marked elements of the matrix in row-major order is ⍵. For instance, assume you have marked a number of numeric fields anywhere in the matrix. To add these values and assign the sum to the field at the cursor position use the command

```
⍺←+/⍵
```

To add 1 to each of the marked elements you may use

```
⍵←⍵+1
```

You should not use functions which display a result on the screen as the location of the output is unpredictable and will be overwritten by UEDIT immediately. To display results which will not become elements of the matrix a utility function *SHOW* is included in the UEDIT workspace. To display the current value of the variable *PATH* simply type

```
SHOW PATH
```

and return to the editor session by hitting the ⟨Return⟩ key. If, for example, you want to add the first ten integers you can type

```
SHOW +/⍳10
```

to see the result.

To display another matrix, say *NEWMATRIX*, on the screen you can give the command

```
USER 'NEWMATRIX'
```

This command calls UEDIT recursively and overlays *NEWMATRIX* over the existing matrix. Note that the name of the matrix must be enclosed within quotes. You can edit *NEWMATRIX* like any other matrix. When you terminate this session by hitting ⟨F3⟩ or ⟨F4⟩, UEDIT takes you back to the original matrix.

As the number of possible commands is nearly unlimited, the only message in case of an error is that UEDIT will display the message "Invalid Input". The command is displayed again on the input line with α and ω expanded to their actual meaning, and you have the chance to correct your input.

For additional examples see Appendix D.

F5 **Statistical Functions**

This option displays a submenu of the available statistical functions. These are described on page 49.

S-F5 **Toggle Column Labels**

Converts the first matrix row to column labels if no column labels exist. Otherwise it adds the existing column labels as a new first row to the matrix. When a new matrix is imported UEDIT guesses whether the first row and first column contain labels or not. If UEDIT's assumption is wrong use this function or "Toggle Row Labels" ⟨S-F6⟩ to correct the mistake.

C-F5 **Edit Column Labels**

This function lets you edit existing column labels or create new labels if none exist. It works in bulk mode (see the next function) starting with the first column. When all desired changes are made you may stop by hitting the "Escape" key.

F6 **Bulk mode**

The bulk mode option allows the user to manually add or insert new rows or columns into the current matrix. UEDIT first prompts for row- or column-wise input. For row-wise input the following actions take place (the equivalent holds for column-wise input): A new row is inserted into the matrix before the current cursor position, the cursor is located in the first field of this row and a prompt is displayed to enter a value for this field. Each time you hit ⟨Enter⟩ the cursor changes to the next field to the right and again waits for input. If the row is filled a new row is created below the last one and the process starts over. Hit the ⟨Escape⟩ key to leave this mode.

Note that data which exceed the current column widths will appear truncated on the screen during the input. This improves the speed with which UEDIT can handle the input and stops the input flow from being interrupted. After completion of the input the necessary column widths are recalculated and the display is updated.

S-F6 **Toggle Row Labels**
Converts the first matrix column to row labels if no row labels exist. Otherwise it adds the existing row labels as a new first column to the matrix. When a new matrix is imported UEDIT guesses whether the first column and first row contain labels or not. If UEDIT's assumption is wrong use this function or "Toggle column Labels" ⟨S-F5⟩ to correct the mistake.

F7 **Insert Row**
Inserts a new empty row before the current cursor location.

S-F7 **Insert Column**
Inserts one or more empty columns before the current cursor location. You will be prompted for the formats of the new column. See page 38 for a description of valid formats.

C-F7 **Change Column Format**
Lets you change the format of the column the cursor is currently located on. You will be prompted to specify the new format. See page 38 for a description of the format codes. Note that only valid changes will be accepted. For example, a numeric column can always be changed to character format, but a character-type column can only be converted to numbers if all fields can be interpreted as numbers.

F8 **Search**
This function searches for the next occurence of a specified number or character string. Substrings (even in numbers) will be found too. The search, which is case-sensitive, is performed in row-major order starting at the current cursor position. The function does not "wrap around" the end of the matrix. Therefore, in order to locate all occurrences of the search object you should start in the upper left corner of the matrix.

S-F8 **Recode**
This function is especially useful when you have imported data from a system which can handle only numeric data, thus requiring that character type attributes are coded with numbers. With this function it is easy to recode such a matrix column to its original or any other desired attribute. UEDIT will display each distinct value of that column and prompt for a new attribute. If you want to change only a few values you can terminate this function with ⟨Escape⟩ after all necessary changes have been made.

C-F8 **Rotate**
To edit a matrix it may sometimes be easier to transpose the matrix, i.e., turn rows into columns and columns into rows, and in addition hide matrix areas you do not need. "Rotate" creates a new matrix containing the transpose of the original matrix if no area is marked or of a marked area of the matrix. The new matrix is overlaid over its "parent" matrix and can be edited in the same way.

To return to the original matrix use "Quit" ⟨F3⟩ or "File" ⟨F4⟩. With ⟨F4⟩ all changes are entered into the parent matrix.

F9 **Copy**
Copies a marked row or column block to a new position which is indicated by the current cursor position. Note that the block is inserted before the current row or column. Presently this function will only copy blocks which cover all rows or all columns. To copy smaller areas use the "Put/Get" combination ⟨S-F3⟩, ⟨C-F3⟩.

S-F9 **Move**
Moves a marked row or column block to a new position which is indicated by the current cursor position. This is essentially the same operation as "Copy" but the marked block will be deleted from its original position.

C-F9 **Change "Missing Value"**
This option provides an easy way to change the numeric "Missing Value", i.e., the code assigned to unknown numeric data. The display will be updated immediately after the change.

F10 **Delete Row(s)**
This function deletes marked rows or the current row if no rows are marked. You will be prompted for confirmation before any action takes place. If you have deleted rows by mistake you can still recover from that error if you have not saved the matrix since the deletion. Take the following steps to save as much of your work as possible:

1. Assuming you are editing the matrix *MATRIX*, choose function ⟨C-F4⟩ and issue the APL2 command

   ```
   HELPMATRIX←MATRIX
   ```

   This will copy the original matrix which still contains the deleted rows to a new matrix.

2. "File" the current matrix with the ⟨F4⟩ function.

3. Now edit *HELPMATRIX* and delete all rows except those you want to recover. "File" this matrix.

4. Restart your editor session of *MATRIX*, locate the cursor on the row where the deleted rows should be and insert *HELPMATRIX* using the "Get" function ⟨C-F3⟩.

S-F10 **Delete Colum(s)**
Deletes marked columns or the current column if no columns are marked. You will be prompted for confirmation before any action takes place. See the function ⟨F10⟩ for recommendations for error recovery.

C-F10 **Printer Functions**
Displays a submenu of available printer functions. See page 54 for a description.

K. STATISTICAL FUNCTIONS

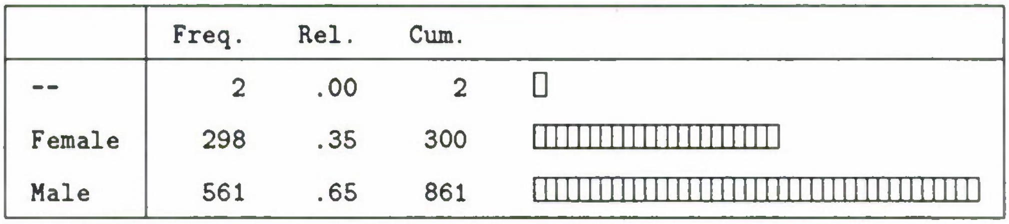
Presently there are four statistical functions implemented into the menuing system of UEDIT. All operate on one or more columns of the currently edited matrix. Function key ⟨F5⟩ activates a submenu which lets you choose from the functions which are described in the following sections.

1. Frequency Counts with Conditionals

UEDIT prompts you for a column number on which to perform the frequency tabulation. If the column contains numeric data or dates you also have to specify three classification parameters: lower bound, upper bound and number of classes (see page 52 for details). To include only certain observations into the frequency count or to exclude certain observations you have the option of conditioning the tabulation on one or more matrix columns, including the one which is counted, as described on page 52. For example, you may want to exclude from the count the cells marked unknown.

The function then creates a new matrix overlaid over its parent in which each row contains the class label, absolute, relative and cumulative frequencies. The last matrix column displays a simple bar chart to visualize the frequencies. If all absolute frequencies are smaller than 40 then all bar lengths (measured in display columns) are equal to those frequencies. Otherwise the longest bar will be 40 columns long and the others have lengths proportional to it. The lengths are at least 1 unless there are no observations in a class.

You can edit the table like any other matrix. When you decide to sort the rows on a different criterion than the default lexicographical order the cumulative frequencies are recalculated automatically. To go back to the original table hit ⟨F4⟩ (File) to save the table to disk or ⟨F3⟩ (Quit) to exit without saving the table.



```
          Freq.   Rel.   Cum.

--            2    .00      2
Female      298    .35    300
Male        561    .65    861
```

Figure 2. Sample frequency table created by UEDIT

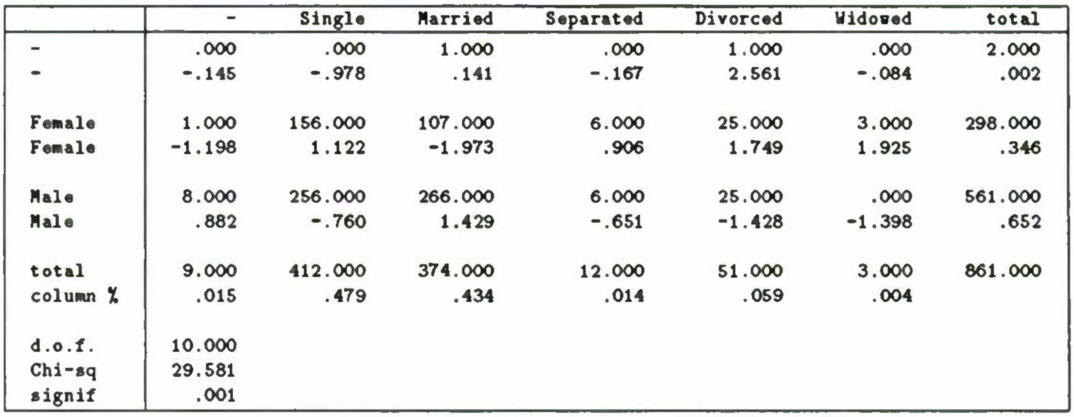
2. Crosstabulation with Conditionals

This option allows the user to crosstabulate any two matrix columns. Thus you will have to specify two columns at UEDIT's prompt. The handling of numeric columns and the conditioning are the same as in the case of frequency counts.

Again UEDIT creates a new table which is overlaid over the original matrix. It contains the observed absolute frequencies and the standard residuals for each field of the table. The standard residuals are highlighted at color level 1. In addition, residuals whose absolute values are larger than 1.96 or 2.54 are highlighted to level 2 and level 3, respectively, for emphasis. Thus, to see only the standard residuals, you can use the Shadow function ⟨Shift-S⟩ to hide the other rows and columns.

Also displayed are the row and column marginals in absolute and relative numbers. Below the table the value of the χ^2 statistic, the p-value and the number of degrees of freedom are tabulated.

To return to the parent matrix hit ⟨F3⟩ or ⟨F4⟩ as always.



```
           |       -    Single   Married  Separated  Divorced   Widowed     total
-          |    .000      .000     1.000       .000     1.000      .000     2.000
-          |   -.145     -.978      .141      -.167     2.561     -.084      .002

Female     |   1.000   156.000   107.000      6.000    25.000     3.000   298.000
Female     |  -1.198     1.122    -1.973       .906     1.749     1.925      .346

Male       |   8.000   256.000   266.000      6.000    25.000      .000   561.000
Male       |    .882     -.760     1.429      -.651    -1.428    -1.398      .652

total      |   9.000   412.000   374.000     12.000    51.000     3.000   861.000
column %   |    .015      .479      .434       .014      .059      .004

d.o.f.     |  10.000
Chi-sq     |  29.581
signif     |    .001
```

Figure 3. Sample contingency table created by UEDIT

3. Draftsman's Display

This function creates 2–way contingency tables for several matrix columns which are laid out internally as

$$\begin{array}{ccccc}
V_1 \text{ vs. } V_2 & V_1 \text{ vs. } V_3 & V_1 \text{ vs. } V_4 & \ldots & V_1 \text{ vs. } V_n \\
 & V_2 \text{ vs. } V_3 & V_2 \text{ vs. } V_4 & \ldots & V_2 \text{ vs. } V_n \\
 & & V_3 \text{ vs. } V_4 & \ldots & V_3 \text{ vs. } V_n \\
 & & & \ddots & \vdots \\
 & & & & V_{n-1} \text{ vs. } V_n
\end{array}$$

where V_i vs. V_j denotes the result of a crosstabulation of columns i and j.

Each table is formatted as the single tables described in the previous subsection, and you can edit each table in any way. To switch to a different submatrix hold down the ⟨Alt⟩-key and hit one of the cursor keys. Note that you must release ⟨Alt⟩ to start the scrolling.

UEDIT chooses a temporary name for each table, which is a composite of the corresponding column labels or column numbers if no labels exist. Recall, that the name of a matrix is always displayed in the bottom row of the screen.

4. Aggregation

To increase cell counts and cell expectations in contingency tables you can aggregate ("pool") matrix columns or rows. Simply enter the column or row numbers in response to UEDIT's prompt or answer "O" which returns you to the original contingency table. All necessary recalculations will be performed automatically. Note that when you want to aggregate rows, you have to enter the row numbers which are displayed in the left-most column of the display. It is not necessary that the columns or rows you aggregate are contiguous.

When you are in a draftsman's display the pooling takes place only in a particular table, i.e., the other tables are not changed and can be aggregated in a total different way.

5. Classification of Numeric Data

If you want to do frequency counts or crosstabulations on numeric columns or on columns formatted as dates UEDIT assumes that the values are from a continuous domain. You have to enter three classification parameters:

1. the lower limit L of the first class,
2. the upper limit U of the last class and
3. the number of classes n in which to group the data.

The classes then have equal lengths $l = (U - L)/n$. Two additional classes are created to classify values below L and above U. That means $n + 2$ classes are created

$$(-\infty, L_1),\ [L_1, U_1),\ \ldots,\ [L_n, U_n),\ [U_n, +\infty)$$

where $U_i = L + i \times l$ for $i = 1, \ldots, n$ and $L_1 = L$, $L_j = U_{j-1}$ for $j = 2, \ldots, n$. The intervals are open to the right, except for the first interval.

If the data have only a few different values (e.g., re-coded character labels) it is suggested that the user converts this column to character data ⟨Ctrl-F7⟩ before starting the function to emphasize their "discrete" status.

6. Conditional Calculations

UEDIT allows frequency counts and crosstabulations to be conditioned on *conditioning columns*, i.e., to perform the calculations only for those rows that match specific criteria. These conditioning columns can be any matrix columns including the ones which are counted.

The input of conditioning columns is a two-step process: First you enter the column numbers and their logical relationship, then you enter the conditioning criteria for each of the columns successively. A complete example is given at the end of this section.

When UEDIT prompts for conditioning columns hit ⟨Enter⟩ if you want all rows to be included in the operation. Otherwise enter the conditioning columns in a logical expression. For example, to include only those rows in the calculation where both column 1 and column 2 match certain criteria enter

```
1∧2
```

After a syntax check you will be prompted for the criteria for each conditioning column (see below). Valid operators are

```
∧ ∨ ⍲ ⍱ ~ / ( ) ,
```

This makes a construction like

```
1∧(2∨3∧4)∨∧/5,6,7
```

perfectly legal although it may make no statistical sense. The most frequent application will probably be to exclude all missing values from a crosstabulation where you would use the first example, `1∧2`.

The conditioning criteria are inputted in the following way:

1. **Character column**
   Enter the criteria separated by commas. For example, to include only rows where the marital status is single or married type

   ```
   Single,Married
   ```

   The case of your input is significant as are leading blanks. If an entry contains a comma itself enclose it within double quotes (") or diereses (¨) when your keyboard layout is set to APL.

   To exclude certain criteria type a tilde (~) in front of the values followed by a comma, i.e., to exclude rows with a marital status single or married the correct input is

   ```
   ~,Single,Married
   ```

   If missing values are denoted by empty (blank) fields type

   ```
   ~,
   ```

   to exclude them.

2. **Numeric or date column**
   The criteria have the form

   $L_1, U_1, L_2, U_2, \ldots$

   where L_i and U_i define a closed interval $[L_i, U_i]$ specifying the range of values to be included or excluded. For example,

   ```
   7,10,15,20,30,30
   ```

   includes only rows whose value in the conditioning column is in the range $[7,10]$, $[15,20]$ or is exactly $30 = [30,30]$. Overlapping ranges are allowed:

   ```
   12,20,10,15
   ```

   includes all rows with a value in the interval $[12,20] \cup [10,15] = [10,20]$ in the calculation.

As in the case of character columns the tilde excludes certain ranges. To exclude rows with a date between February 1 and March 15, 1990 you have to type

```
~,2-1-90,3-15-90
```

You can omit the year in the input if you type this in 1990. When you want to include the missing values (*MISS∆N*) you can abbreviate this to

```
~,
```

To summarize, assume you want to crosstabulate columns 3 and 6 of a matrix but want to exclude the missing values in both columns. Then your answers to UEDIT's prompts would be:

(Columns to crosstabulate:)	3,6
(Conditioning columns:)	3∧6
(Criteria for column 3:)	~,
(Criteria for column 6:)	~,

L. PRINTER FUNCTIONS

The ⟨Ctrl-F10⟩ key combination activates a submenu with several options. These are described in the following subsections. If you use a Hewlett Packard LaserJet II or compatible printer it is recommended that you call the Auxiliary Printer Processor AP81 on the DOS command line instead of the AP80 supplied by IBM. Note that you can always take a "snapshot" of the current screen without using the special printer functions by hitting ⟨Shift-PrtScrn⟩.

F1 **Print worksheet**
Prints the complete matrix with page numbers and headings. Matrices which do not fit on a single page are split on several continuing pages. Columns are not broken over pages. The page numbering is done in the following layout:

1.1	1.2	...
2.1	2.2	...
...	...	...

You are given the option to repeat column and row labels on each page.

F2 **Print mark area**
Works in the same way as "Print worksheet" but prints only a marked area of the currently edited matrix (see page 41 for a description of "marked area"). This feature is very useful and gives the user great control over the printed output.

F3 **Formfeed**
Sends a formfeed to the printer, i.e., ejects a page.

F4 **Initialize**
This function initializes the printer. The only task presently is (in the case of the AP81) to download a portrait font `AP100RFN.SFP` and a landscape font `AP100RFN.SFL` to the memory of the laser printer. You can use any fonts provided they have these names (or are renamed to them) and reside in the default directory of the default disk. UEDIT displays a warning message if it cannot found one or both fonts but does not take any further action. You have to initialize the printer only once; the fonts remain in memory until you turn off the printer.

F5 **Orientation**
Switches between portrait mode (the default) and landscape orientation. The function also exchanges the valus for textheight and textwidth against each other. The menu always shows the mode you switch to when you choose this option. I.e., when you read "portrait" on the menu screen you are currently in landscape mode.

F6 **Left margin**
Specifies the blank space (in printer columns) to the left of the printer matrix.

F7 **Number of columns**
Sets the number of columns to be printed per page. Note that this number specifies the actual printer positions, usually the number of characters. It is not related to the columns of the matrix.

F8 **Top margin**
Defines the number of blank lines above the page number, which is the first line printed.

F9 **Lines per inch**
Sets the vertical spacing of the printout. The most used values are 6 and 8 lines per inch. Depending on the size of the font and the size of the worksheet to be printed you may increase or decrease the value.

F10 **Reset**
This option resets all values set with functions ⟨F5⟩–⟨F9⟩ to their default values which are saved in the global variable *PRINT*.

The global variable *PRINT* is a 7-element vector containing the following default values

PRINT[1]	11	PH	paper height
PRINT[2]	8.5	PW	paper width
PRINT[3]	0		flag portrait(0) / landscape(1)
PRINT[4]	10	LM	left margin (in printer columns)
PRINT[5]	80	PC	number of columns to print
PRINT[6]	3	TM	top margin (in lines)
PRINT[7]	6	LPI	lines per inch

For example, to make landscape printing the default mode change *PRINT* by assigning

```
PRINT[⍳3]←8.5 11 1
```

From these parameters other necessary values can be calculated. The number of lines printed per page (textheight *TH*) is defined by

$$TH = \text{integer}\left(LPI \times \left(PH - \frac{TM}{LPI} - \frac{1}{2}\right)\right)$$

The "usable" textheight is TH minus 3 lines for the page number and the worksheet title minus 2 lines if column labels are printed.

To calculate horizontal margins in inches the "pitch" of the font must be known. The pitch is the number of characters per horizontal inch. Usual pitch values for dot matrix printers are 10 or 12. The laser printer fonts `AP100RFN.SFP` and `AP100RFN.SFL` both print 12 characters per inch. Then the left margin LMI and right margin RMI in inches are defined by

$$\begin{aligned} LMI &= \frac{LM}{\text{pitch}} \\ RMI &= PW - \frac{LM + NC}{\text{pitch}} \end{aligned}$$

To obtain a left margin of l and a right margin of r inches set

$$\begin{aligned} LM &= l \times \text{pitch} \\ NC &= (PW - r) \times \text{pitch} - LM \end{aligned}$$

M. CLEANING UP

When you finish your editing session you may want to save global variables you have created in a separate file or erase these objects to clean up the workspace before you start a new session. The function *CLEAN* helps you with these tasks. Simply type

```
    CLEAN
```

on the APL2 command line. You have are given choices:

1. To erase all functions and variables which are part of the UEDIT system,
2. To erase all objects which are <u>not</u> part of the UEDIT system.

Respond by typing ⟨1⟩ or ⟨2⟩ followed by ⟨Return⟩ depending on your choice. Any other key combination will cancel the execution of this function.

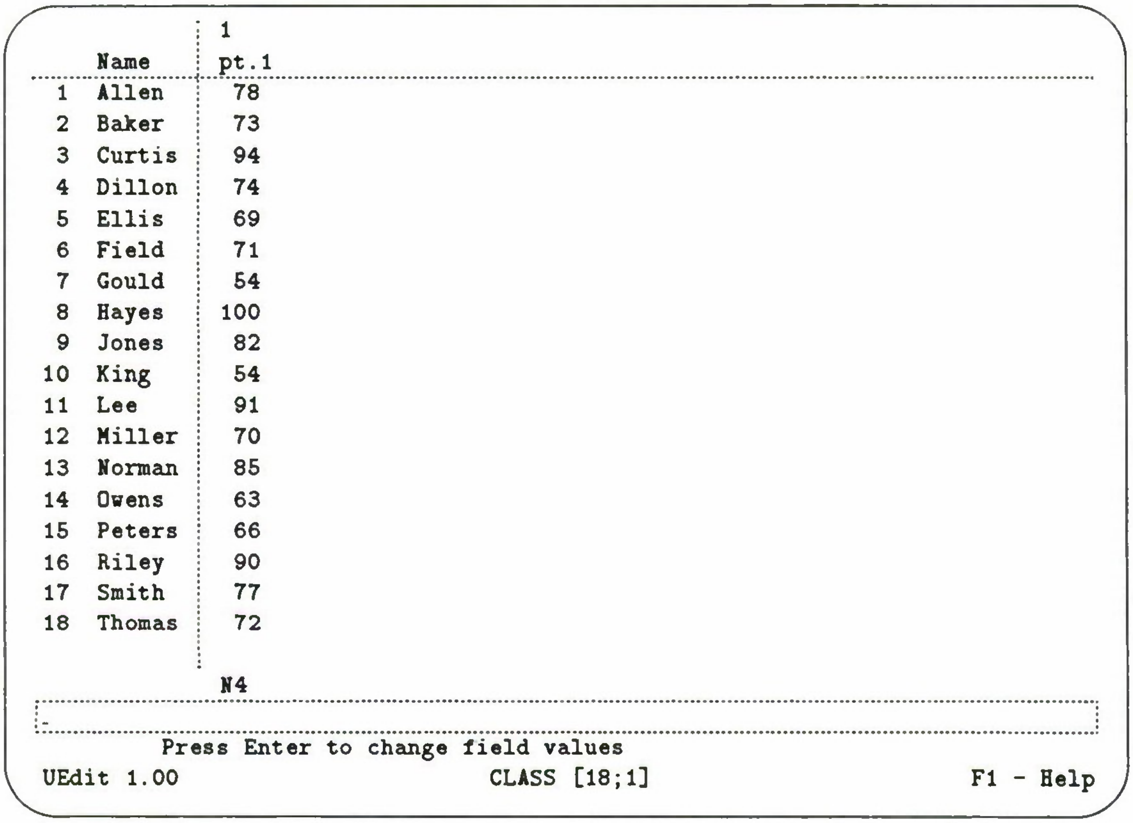
If you want additional functions or variables to be recognized by the *CLEAN* function as an integral part of UEDIT, copy them into the workspace, give the commands

```
    ⍙nl2←Dtb¨⊂[2]⎕NL 2
    ⍙nl3←Dtb¨⊂[2]⎕NL 3
```

and save the UEDIT workspace.

APPENDIX D. SAMPLE SESSION WITH UEDIT

The following sample session tries to make the user more familiar with some of UEDIT's features. Professor User of the Naval Postgraduate School currently teaches a class with 18 students. The grading for the course is based on two examinations which determine 40% and 60% of the final grade, respectively. At the beginning of the quarter he has prepared a UEDIT matrix *CLASS* with the names of the students which he uses as row labels. After the first exam he enters the points for each student with UEDIT's bulk mode ⟨F6⟩. This allows him to enter the scores one after the other without looking up from his notes. The result after adding column labels with ⟨Ctrl-F5⟩ is a display as shown in Figure 4.



```
                   : 1
        Name       : pt.1
...................:.............................................................
   1    Allen      :  78
   2    Baker      :  73
   3    Curtis     :  94
   4    Dillon     :  74
   5    Ellis      :  69
   6    Field      :  71
   7    Gould      :  54
   8    Hayes      : 100
   9    Jones      :  82
  10    King       :  54
  11    Lee        :  91
  12    Miller     :  70
  13    Norman     :  85
  14    Owens      :  63
  15    Peters     :  66
  16    Riley      :  90
  17    Smith      :  77
  18    Thomas     :  72
                   :
                     N4
:...............................................................................:
:-                                                                              :
:...............................................................................:
          Press Enter to change field values
 UEdit 1.00                        CLASS [18;1]                       F1 - Help
```

Figure 4. UEDIT display after input of the results of the first exam

Professor User has a standard scheme to translate points, which always have 100 as a maximum, into grades:

points	94–100	87–93	80–86	73–79	66–72	59–65	52–58
grade	A	A−	B+	B	B−	C+	C

Therefore he has written a short APL2 function *GRADE* which allows him to do the conversion efficiently:

```
[0] R←GRADE PT
[1] R←('A' 'A-' 'B+' 'B' 'B-' 'C+' 'C')[⌊(107-PT)÷7]
```

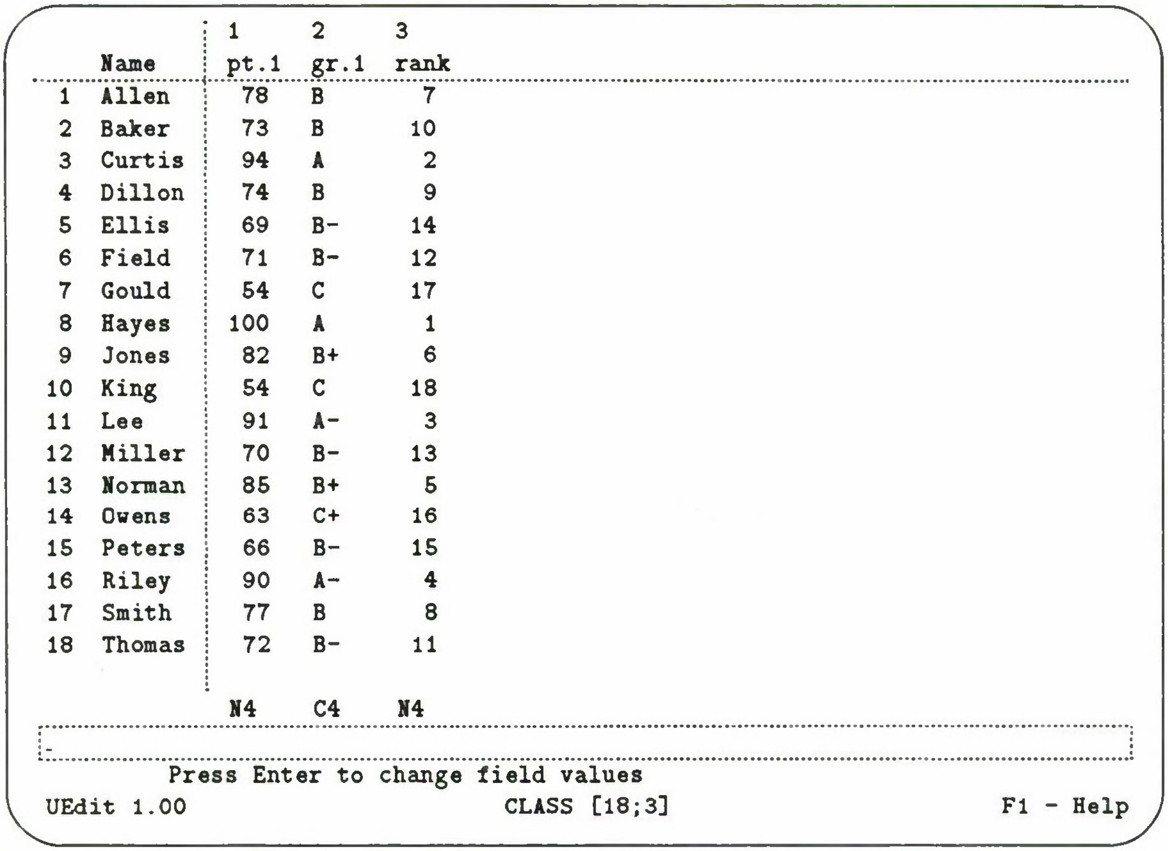
He is also interested in the ranking of the students. So he decides to add two columns to the matrix. He moves the cursor to the right of the "points" column, hits ⟨S-F7⟩ to insert columns and answers the prompt for column types with

```
C,A
```

because the grades have character type but the ranks are numbers. Both tasks take column 1 as input. So he marks column 1 by moving the cursor into it and hitting ⟨Ctrl-C⟩. This allows him to use ⍵ as a short-cut notation for *MAT*[;1]. Then he uses UEDIT's feature to enter arbitrary APL2 commands (⟨Ctrl-F4⟩) twice and assigns, on the command line,

```
MAT[;2]←GRADE ⍵
MAT[;3]←⍋⍒⍵
```

This inserts the grades into column 2 and the ranks into column 3 as shown in Figure 5. He unmarks column 1 (⟨Ctrl-U⟩) and files the matrix with ⟨F4⟩.



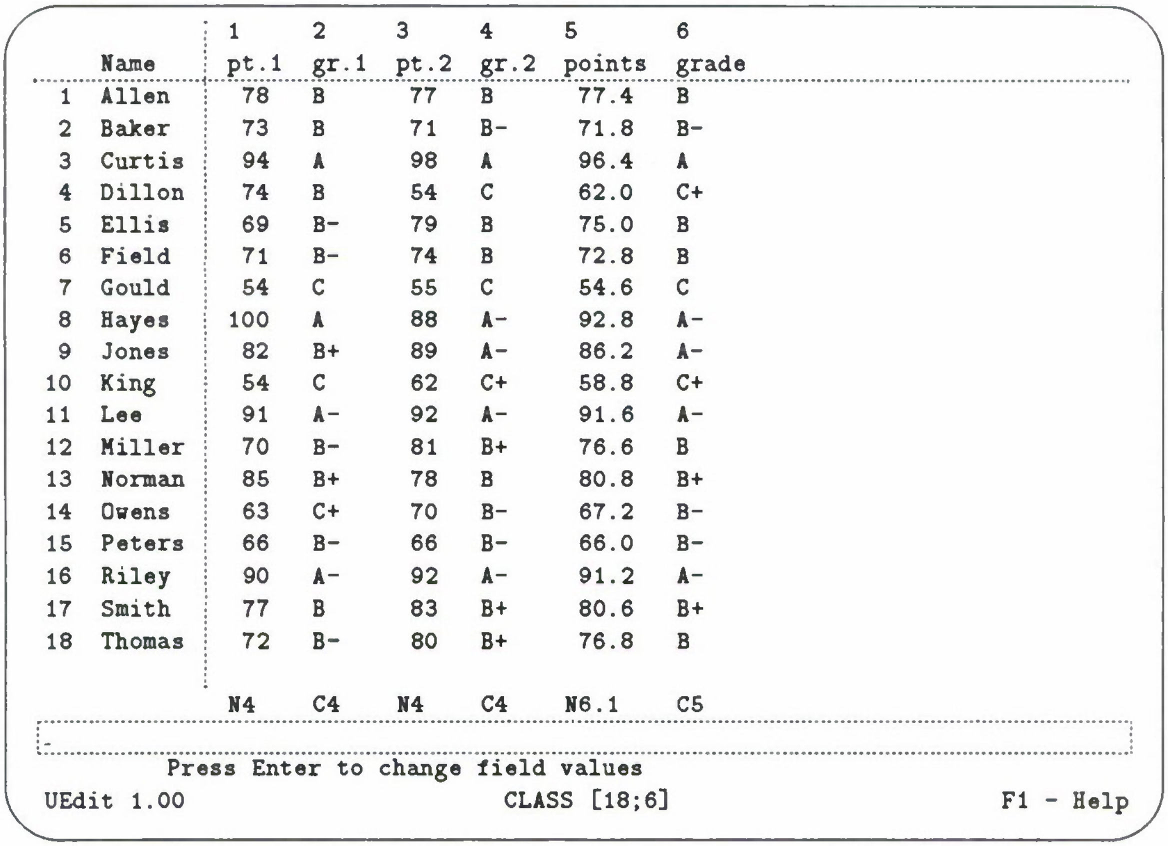
```
                 :  1      2      3
        Name     :  pt.1   gr.1   rank
.......................................................................
   1   Allen     :   78    B         7
   2   Baker     :   73    B        10
   3   Curtis    :   94    A         2
   4   Dillon    :   74    B         9
   5   Ellis     :   69    B-       14
   6   Field     :   71    B-       12
   7   Gould     :   54    C        17
   8   Hayes     :  100    A         1
   9   Jones     :   82    B+        6
  10   King      :   54    C        18
  11   Lee       :   91    A-        3
  12   Miller    :   70    B-       13
  13   Norman    :   85    B+        5
  14   Owens     :   63    C+       16
  15   Peters    :   66    B-       15
  16   Riley     :   90    A-        4
  17   Smith     :   77    B         8
  18   Thomas    :   72    B-       11
                 :
                    N4     C4     N4
 .....................................................................
 :-                                                                  :
 .....................................................................
           Press Enter to change field values
 UEdit 1.00                      CLASS [18;3]                F1 - Help
```

Figure 5.   UEDIT display after adding grades and ranks

After the second examination, Professor User deletes the ranking column with function ⟨S-F10⟩. Again he uses the bulk mode to enter the points for the second exam and then converts them to grades with his *GRADE* function. He adds two more columns to the matrix, which will contain the course points and course grade, and fills these columns with the APL2 command

```
MAT[;6]←GRADE MAT[;5]←MAT[;1 3]+.×.4 .6
```

This gives him the display of Figure 6.



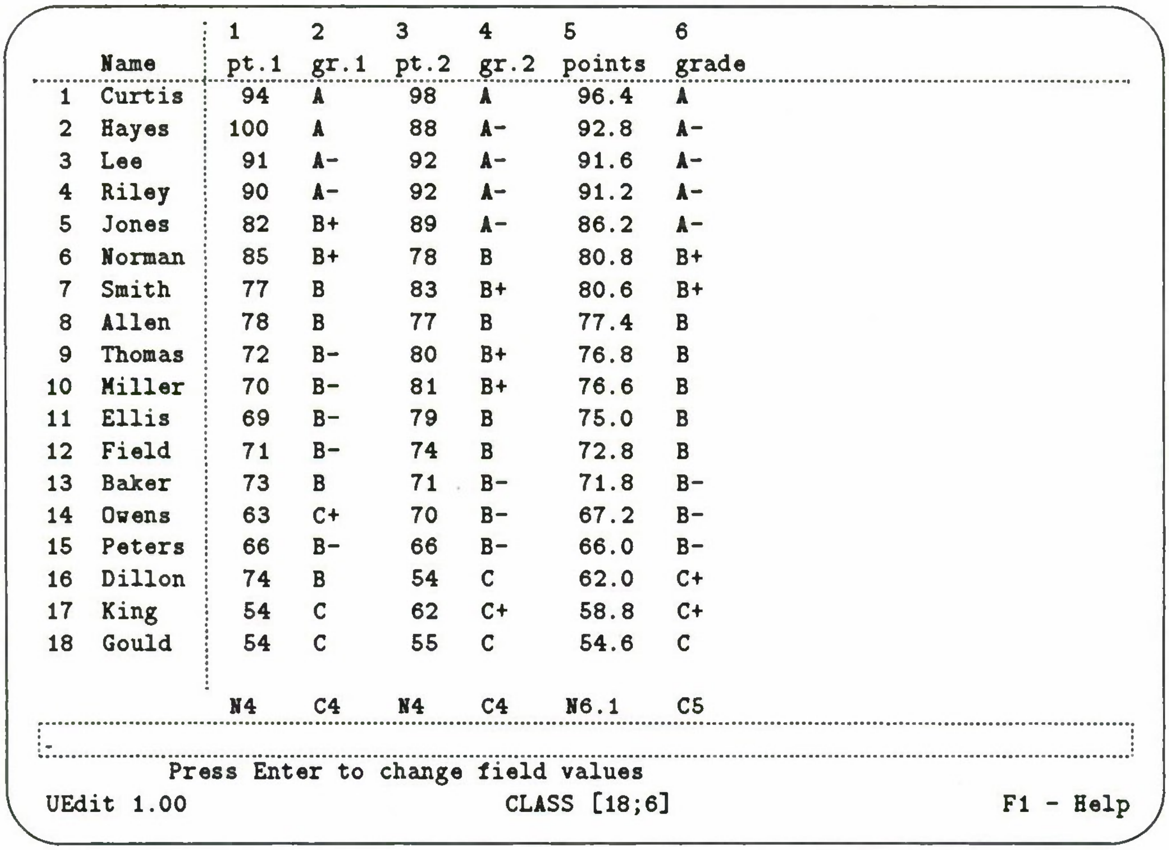
```
                  : 1      2      3      4      5        6
          Name    : pt.1   gr.1   pt.2   gr.2   points   grade
..................:.............................................................
      1  Allen    :  78    B       77    B       77.4    B
      2  Baker    :  73    B       71    B-      71.8    B-
      3  Curtis   :  94    A       98    A       96.4    A
      4  Dillon   :  74    B       54    C       62.0    C+
      5  Ellis    :  69    B-      79    B       75.0    B
      6  Field    :  71    B-      74    B       72.8    B
      7  Gould    :  54    C       55    C       54.6    C
      8  Hayes    : 100    A       88    A-      92.8    A-
      9  Jones    :  82    B+      89    A-      86.2    A-
     10  King     :  54    C       62    C+      58.8    C+
     11  Lee      :  91    A-      92    A-      91.6    A-
     12  Miller   :  70    B-      81    B+      76.6    B
     13  Norman   :  85    B+      78    B       80.8    B+
     14  Owens    :  63    C+      70    B-      67.2    B-
     15  Peters   :  66    B-      66    B-      66.0    B-
     16  Riley    :  90    A-      92    A-      91.2    A-
     17  Smith    :  77    B       83    B+      80.6    B+
     18  Thomas   :  72    B-      80    B+      76.8    B
                  :
                    N4     C4     N4     C4     N6.1     C5

           Press Enter to change field values
 UEdit 1.00                    CLASS [18;6]                    F1 - Help
```

Figure 6. UEDIT display after calculating the final scores

To rank the students he chooses to sort the rows of the matrix according to the points of column 5. In the case of equal numbers he wants the points of the second examination (column 3) to be the criterion for a higher rank. Therefore he starts the sorting with ⟨S-F1⟩ and responds to UEDIT's prompt for column numbers

```
¯5,¯3
```

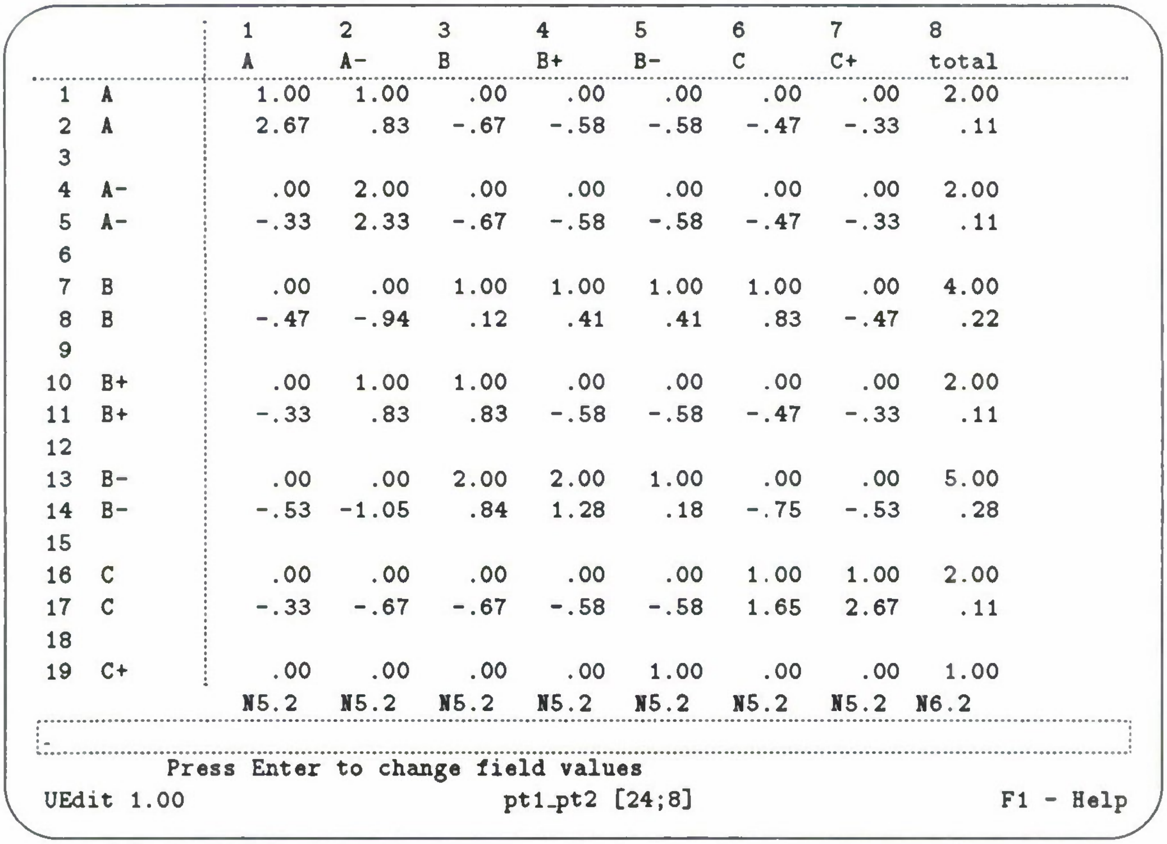
This will show him the students in descending order of their points, i.e., the best student's name is listed in the first row, as shown in Figure 7.



```
                 : 1     2     3     4     5        6
        Name     : pt.1  gr.1  pt.2  gr.2  points   grade
.........................................................................
    1   Curtis   :  94   A      98   A      96.4    A
    2   Hayes    : 100   A      88   A-     92.8    A-
    3   Lee      :  91   A-     92   A-     91.6    A-
    4   Riley    :  90   A-     92   A-     91.2    A-
    5   Jones    :  82   B+     89   A-     86.2    A-
    6   Norman   :  85   B+     78   B      80.8    B+
    7   Smith    :  77   B      83   B+     80.6    B+
    8   Allen    :  78   B      77   B      77.4    B
    9   Thomas   :  72   B-     80   B+     76.8    B
   10   Miller   :  70   B-     81   B+     76.6    B
   11   Ellis    :  69   B-     79   B      75.0    B
   12   Field    :  71   B-     74   B      72.8    B
   13   Baker    :  73   B      71   B-     71.8    B-
   14   Owens    :  63   C+     70   B-     67.2    B-
   15   Peters   :  66   B-     66   B-     66.0    B-
   16   Dillon   :  74   B      54   C      62.0    C+
   17   King     :  54   C      62   C+     58.8    C+
   18   Gould    :  54   C      55   C      54.6    C
                 :
                   N4    C4    N4    C4    N6.1     C5
 ........................................................................
 :-                                                                     :
 ........................................................................
              Press Enter to change field values
  UEdit 1.00                       CLASS [18;6]              F1 - Help
```

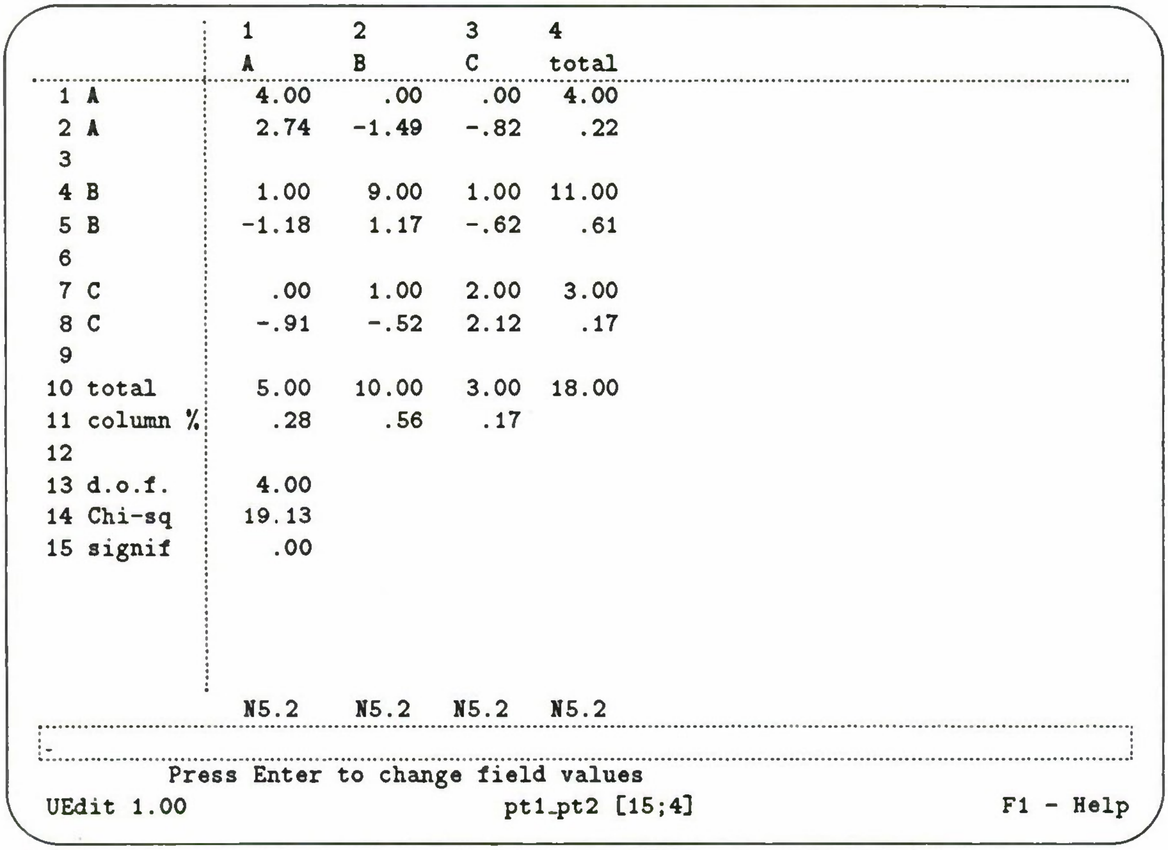
Figure 7. UEDIT display after sorting on final scores

Professor User then decides to crosstabulate the grades of the two exams although he knows that this will not make much sense as there are only 18 students in his class. But he wants to become more familiar with UEDIT and uses every opportunity to gain experience with its features. So he hits ⟨F5⟩ to open the menu of statistical functions, chooses ⟨F2⟩ for crosstabulation and enters 2,4 as the columns of interest. The result is a 7 × 7 contingency table, shown in Figure 8. Of course, most of the observed frequencies are 0. He decides to aggregate the scores into only three classes A, B and C by pooling (A,A-), (B,B+,B-) and (C,C+). He does this for the rows as well as for the columns of his matrix. Note that, when asked for row numbers to aggregate, he responds with "real" matrix rows, that is for example, rows 1 and 4 to pool grades A and A-. Figure 9 shows the resulting 3 × 3 contingency table.



```
          :  1       2       3       4       5       6       7       8
          :  A       A-      B       B+      B-      C       C+      total
..........:.....................................................................
  1  A    :   1.00    1.00     .00     .00     .00     .00     .00    2.00
  2  A    :   2.67     .83    -.67    -.58    -.58    -.47    -.33     .11
  3       :
  4  A-   :    .00    2.00     .00     .00     .00     .00     .00    2.00
  5  A-   :   -.33    2.33    -.67    -.58    -.58    -.47    -.33     .11
  6       :
  7  B    :    .00     .00    1.00    1.00    1.00    1.00     .00    4.00
  8  B    :   -.47    -.94     .12     .41     .41     .83    -.47     .22
  9       :
 10  B+   :    .00    1.00    1.00     .00     .00     .00     .00    2.00
 11  B+   :   -.33     .83     .83    -.58    -.58    -.47    -.33     .11
 12       :
 13  B-   :    .00     .00    2.00    2.00    1.00     .00     .00    5.00
 14  B-   :   -.53   -1.05     .84    1.28     .18    -.75    -.53     .28
 15       :
 16  C    :    .00     .00     .00     .00     .00    1.00    1.00    2.00
 17  C    :   -.33    -.67    -.67    -.58    -.58    1.65    2.67     .11
 18       :
 19  C+   :    .00     .00     .00     .00    1.00     .00     .00    1.00
             N5.2    N5.2    N5.2    N5.2    N5.2    N5.2    N5.2   N6.2
.................................................................................
:-                                                                              :
.................................................................................
            Press Enter to change field values
 UEdit 1.00                     pt1_pt2 [24;8]                        F1 - Help
```

Figure 8. UEDIT display of the contingency table



```
             :  1       2       3     4
             :  A       B       C     total
.............:...........................................................
  1 A        :  4.00     .00     .00   4.00
  2 A        :  2.74   -1.49    -.82    .22
  3          :
  4 B        :  1.00    9.00    1.00  11.00
  5 B        : -1.18    1.17    -.62    .61
  6          :
  7 C        :   .00    1.00    2.00   3.00
  8 C        :  -.91    -.52    2.12    .17
  9          :
 10 total    :  5.00   10.00    3.00  18.00
 11 column % :   .28     .56     .17
 12          :
 13 d.o.f.   :  4.00
 14 Chi-sq   : 19.13
 15 signif   :   .00
             :
             :
             :
             :
                N5.2    N5.2    N5.2  N5.2
:........................................................................:
:-                                                                       :
:........................................................................:
          Press Enter to change field values
UEdit 1.00                        pt1_pt2 [15;4]                 F1 - Help
```

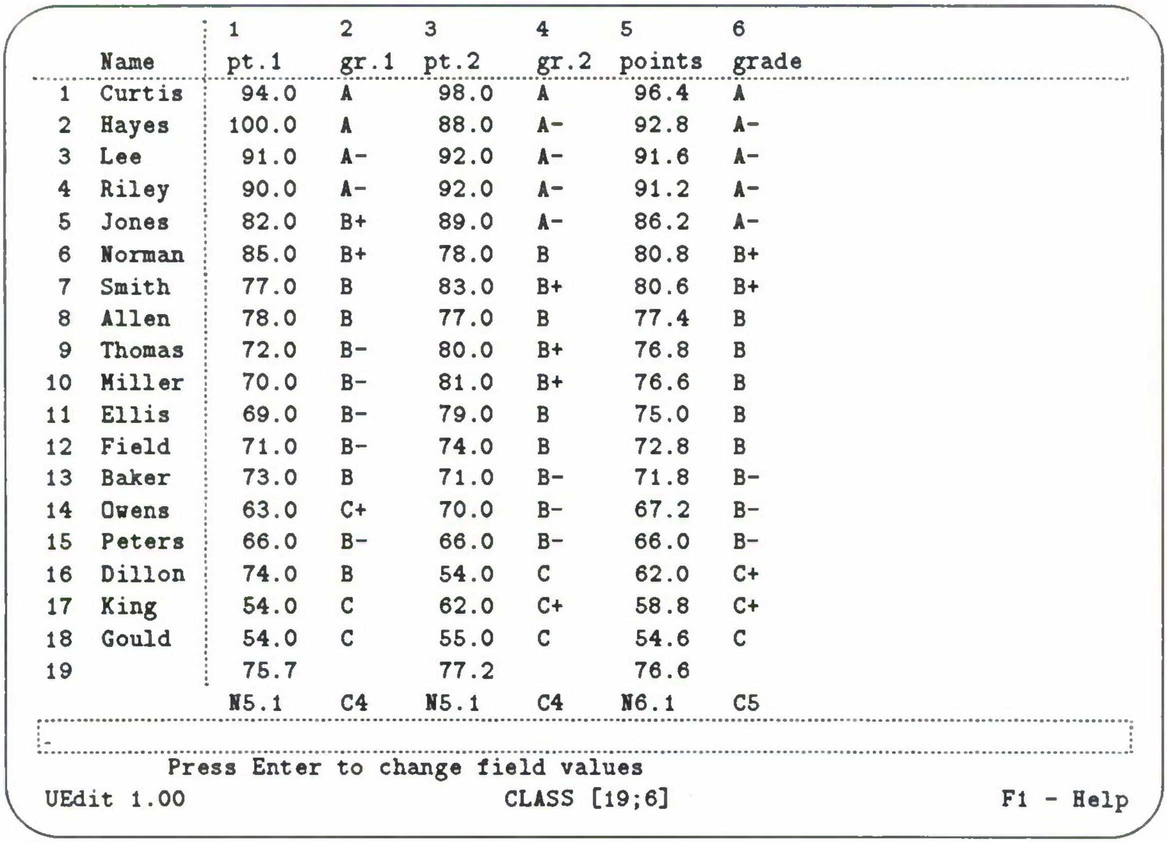
Figure 9. UEDIT display of the contingency table after aggregation

After this excursion Professor User leaves the contingency table display by hitting ⟨F3⟩ which takes him back to the original matrix. He wants to know at least one statistical figure, the average scores of his students. He moves the cursor to column 5 and marks it with ⟨Ctrl-C⟩. Then he moves the cursor to the last line, i.e., to the line below the last score, types

```
(+/ω)÷ρω
```

and presses ⟨Enter⟩. Note that he does not have to hit ⟨Enter⟩ to start entering a new value for a field. As soon as he types the left parenthesis, the input line is activated with the parenthesis as first character.

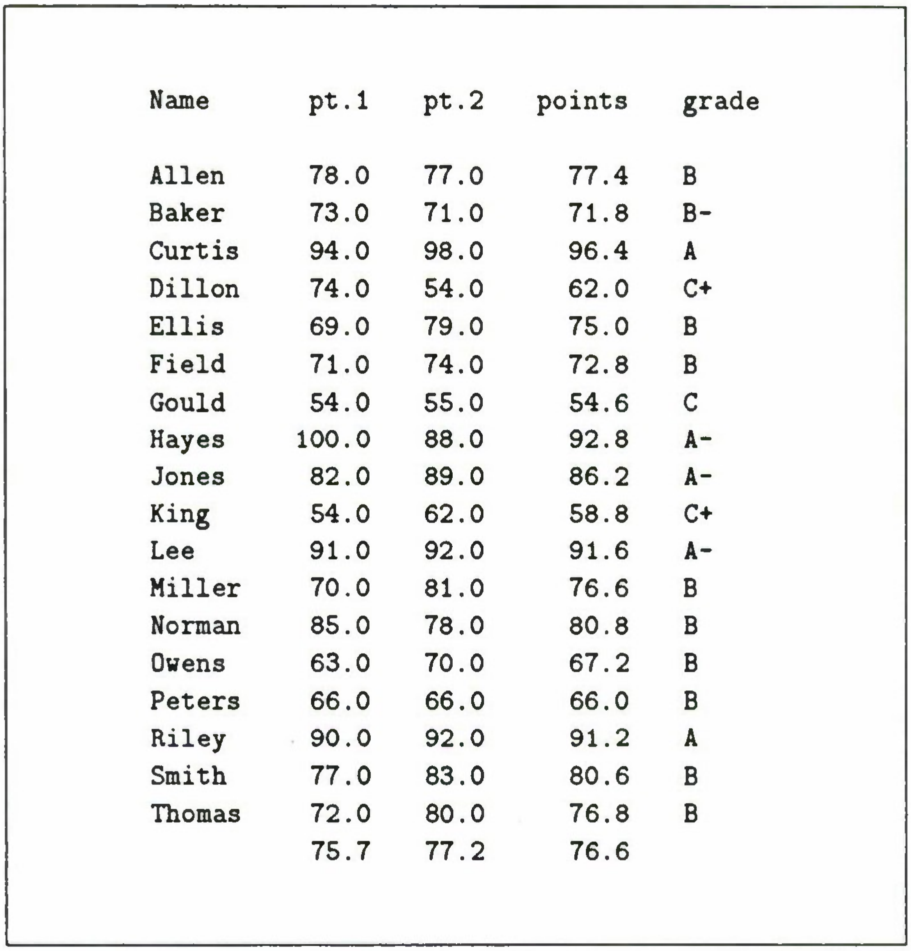
He does the same calculations for columns 1 and 3 and formats these columns so that they are rounded to one decimal ⟨Ctrl-F7⟩. This gives him the display shown in Figure 10.



```
              : 1       2      3        4      5        6
      Name    : pt.1    gr.1   pt.2     gr.2   points   grade
..............:..............................................................
  1   Curtis  :  94.0   A       98.0    A       96.4    A
  2   Hayes   : 100.0   A       88.0    A-      92.8    A-
  3   Lee     :  91.0   A-      92.0    A-      91.6    A-
  4   Riley   :  90.0   A-      92.0    A-      91.2    A-
  5   Jones   :  82.0   B+      89.0    A-      86.2    A-
  6   Norman  :  85.0   B+      78.0    B       80.8    B+
  7   Smith   :  77.0   B       83.0    B+      80.6    B+
  8   Allen   :  78.0   B       77.0    B       77.4    B
  9   Thomas  :  72.0   B-      80.0    B+      76.8    B
 10   Miller  :  70.0   B-      81.0    B+      76.6    B
 11   Ellis   :  69.0   B-      79.0    B       75.0    B
 12   Field   :  71.0   B-      74.0    B       72.8    B
 13   Baker   :  73.0   B       71.0    B-      71.8    B-
 14   Owens   :  63.0   C+      70.0    B-      67.2    B-
 15   Peters  :  66.0   B-      66.0    B-      66.0    B-
 16   Dillon  :  74.0   B       54.0    C       62.0    C+
 17   King    :  54.0   C       62.0    C+      58.8    C+
 18   Gould   :  54.0   C       55.0    C       54.6    C
 19           :  75.7           77.2            76.6
              : N5.1    C4     N5.1     C4     N6.1     C5
:...........................................................................:
:_                                                                          :
:...........................................................................:
          Press Enter to change field values
UEdit 1.00                         CLASS [19;6]                   F1 - Help
```

Figure 10. UEDIT display after adding average scores

Finally, he wants to print out the scores. He first sorts the rows back to their original order, that is, in ascending order of their names. He does that by responding with a 1 to UEDIT's prompt for the column numbers. For the printout he is not interested in the letter grades of the two exams and wants to omit columns 2 and 4 from the output. Therefore he marks columns 1, 3, 5 and 6 using ⟨Ctrl-C⟩, opens the printer menu with ⟨Ctrl-F10⟩ and chooses option ⟨F2⟩ which will print out only the marked area, in this case the marked columns. The printed output would then look like the one shown in Figure 11.



```
Name       pt.1    pt.2    points    grade

Allen      78.0    77.0      77.4    B
Baker      73.0    71.0      71.8    B-
Curtis     94.0    98.0      96.4    A
Dillon     74.0    54.0      62.0    C+
Ellis      69.0    79.0      75.0    B
Field      71.0    74.0      72.8    B
Gould      54.0    55.0      54.6    C
Hayes     100.0    88.0      92.8    A-
Jones      82.0    89.0      86.2    A-
King       54.0    62.0      58.8    C+
Lee        91.0    92.0      91.6    A-
Miller     70.0    81.0      76.6    B
Norman     85.0    78.0      80.8    B
Owens      63.0    70.0      67.2    B
Peters     66.0    66.0      66.0    B
Riley      90.0    92.0      91.2    A
Smith      77.0    83.0      80.6    B
Thomas     72.0    80.0      76.8    B
           75.7    77.2      76.6
```

Figure 11. UEDIT printout of the final scores

APPENDIX E. UEDIT FUNCTION LISTING

The majority of functions of the UEDIT workspace (Version 1.0) were written by the author. Function listings under subsequent versions will differ. The version number appears in the lower left corner of the display. The functions *CHISQ* and *GAUSS* were originally written by Professor Harold J. Larson of the Naval Postgraduate School for the APL*PLUS system of STSC, Inc. They use fast, numerically stable algorithms and are extremely accurate. Necessary changes for the APL2 system were applied by the author. The functions *∆fv*, *Help* and *Menu* are in a similar form part of the APL2/PC distribution, but were enhanced for the purposes of UEDIT.

The two functions *UEDIT* and *USER* are listed first as they are the most important parts of the workspace. All other functions are called from here. They are listed in alphabetical order.

```
UEDIT ORIG;⎕IO;⍙C;⍙SR;⍙SC;⍙pvar;F;Cs;Ds;KEYB
⍝
⍝  Main Program; initializes parameters, then calls USER, the dispatcher
⍝
 ⎕IO←0
⍝
⍝  Assign undefined keys
 KEYB←(16×256⊥⎕AV⍳⌽2 ⎕PK 0 38)+256⊥⎕AV⍳⌽2 ⎕PK 0 36
 KEYB←3+(16×256⊥⎕AV⍳⌽2 ⎕PK 0,KEYB+3)+256⊥⎕AV⍳⌽2 ⎕PK 0,KEYB+1
 ⎕WA←⎕AV[221]⎕PK 0,KEYB+(58×3)+33                    ⍝ Ctrl-F,  hex DD
 ⎕WA←⎕AV[219]⎕PK 0,KEYB+(58×3)+19                    ⍝ Ctrl-R,  hex DB
 ⎕WA←⎕AV[220]⎕PK 0,KEYB+(58×3)+22                    ⍝ Ctrl-U,  hex DC
⍝
⍝  Initialize AP124, get screen size
 ⎕WA←(⎕EX 2 2ρ'CsDs'),⎕SVR F←2 2ρ'CsDs',0ρ⎕IO←1
 ⎕WA←124 ⎕SVO F
 ⍎(2∨.≠124 ⎕SVO F)/'→0,ρ⎕←''ERROR: AP124 NOT ACTIVE'''
 (⍙SR ⍙SC)←Ds[1;3 4],0ρCs←9
⍝
⍝  Define color settings
⍝   1      - edit area             16 - row/column labels
⍝   2..7   - highlight             17 - type/format line
⍝   8      - command input line    18 - message line
⍝   9      - normal marked         19 - status line
⍝  10..15  - highlight marked      20 - pop-up windows
⍝  Row 1: normal color values
⍝  Row 2: color if active field / blinking
```

```
[ 27]  ΔC←23,28 30 31 26 27 25,7 87,92 94 95 90 91 89,71 71 31 110 110
[ 28]  ΔC←ΔC,[0.5]113,124 126 127 122 123 121,63 117,124 126 127 122 123 121,
          203,0 159 0 0
[ 29] ⍝
[ 30] ⍝  Define standard fields
[ 31]  F←1 6 2,(ΔSC-5),2,ΔC[1;16]                    ⍝  1 - Column labels
[ 32]  F←F,3 1,(ΔSR-6),5 2,ΔC[1;16]                  ⍝  2 - Row labels
[ 33]  F←F,(ΔSR-3),1 1,ΔSC,2,ΔC[1;17]                ⍝  3 - Format line
[ 34]  F←F,ΔSR,1 1,ΔSC,2,ΔC[1;19]                    ⍝  4 - Status line
[ 35]  F←F,(ΔSR-1),1 1,ΔSC,2,ΔC[1;18]                ⍝  5 - Message line
[ 36]  F←F,(ΔSR-2),1 1,ΔSC,2,ΔC[1;8]                 ⍝  6 - User input line
[ 37]  F←F,3 7 19 33 2,ΔC[1;20]                      ⍝  7 - Help screen
[ 38]  F←F,3 6,(ΔSR-6),(ΔSC-5),2,ΔC[1;1]             ⍝  8 - Edit area
[ 39]  F←F,1 1 1 1 2,ΔC[2;20]                        ⍝  9 - cursor position
[ 40]  F←F,1 1 1 5 2,ΔC[2;20]                        ⍝ 10 - upper left corner
[ 41]  F←F,2 6 1 1 2,ΔC[1;16]                        ⍝ 11 - pseudo label
[ 42]  F←F,6 40 13 32 2,ΔC[1;20]                     ⍝ 12 - Submenus
[ 43]  Cs←1,0⍴Ds←12 6⍴F                              ⍝ Initialize fields
[ 44]  Cs←12 0,0⍴Cs←12,0⍴Ds←6 1 1                    ⍝ Cursor position
[ 45] ⍝
[ 46]  ⎕WA←ResPrt                                    ⍝ Default printer parameters
[ 47] ⍝
[ 48]  ⎕WA←USER ORIG                                 ⍝ Start edit session
[ 49] ⍝
[ 50]  →0,F←⎕SVR 2 2⍴'CsDs'                          ⍝ Retract shared variables


[  0] RC←USER ORIG;N;K;L;DISP;ΔLR;ΔLC;ΔLS;MAT;ΔMR;ΔMC;ΔAR;ΔAC;ΔRO;ΔCO;ΔOC;ΔOR;
         PRESS;ΔCL;ΔCT;ΔDM;ΔMP;MSG;MSGO;CFLAG;CLR;CHR;ΔUL;ΔUR;ΔUC;ΔWR;ΔWC;ΔMA;
         ΔDA;MARKC;MARKR;HIGHR;HIGHC;HIGHA;FindTxt;DISA;DISL;DIST;DISB
[  1] ⍝
[  2] ⍝  Main dispatcher for all other functions
[  3] ⍝  RC: 1 - File,  0 - Quit    or flag for draftsmen
[  4] ⍝
[  5]  RC←CFLAG←0                                      ⍝ 'Change' flag
[  6]  FindTxt←''                                      ⍝ 'Search' key
[  7]  ΔUL←'   '                                       ⍝ 'Mark/high' reminder
[  8]  Cs←1 8,0⍴Ds←3 6,(ΔSR-6),(ΔSC-5),2,ΔC[1;1]       ⍝ Pseudo fields
[  9]  Cs←1,9,0⍴Ds←1 1 2 5 0,ΔC[1;16]
[ 10]  ⍎(ORIG≡'')/'→(Menu 2)↓0,NEW'                    ⍝ File menu if no matrix name
[ 11]  →(1+NewMat ORIG)⊃0,REFRESH,RESTART              ⍝ Get the matrix
[ 12] ⍝
[ 13] ⍝  Rebuild screen display
[ 14] ⍝
[ 15] NEW:CFLAG←0                                      ⍝ Initial values
[ 16] ⍝
[ 17] REFRESH:Cs←2,5,0⍴Ds←'     Updating display ...',0⍴Cs←7,5,0⍴Ds←ΔC[2;18]
[ 18]  Refresh
[ 19]  Play
[ 20] RESTART:DISL←VerShift           ⍝ create line numbers, row Labels
[ 21]  DISB←HorShift                  ⍝ create column labels, field types/widths
[ 22]  MSG←MSGO←'     Press Enter to change field values'
[ 23] ⍝
[ 24] Scr0:ClearUL                                    ⍝ clear upper left corner
[ 25]  Cs←1,8,0⍴Ds←ΔUR,ΔUC,ΔWR,ΔWC,2,ΔC[1;1]
```

```
Scr1:DISP←MakeDisP                            ⍝ create edit window
Scr2:DISA←MakeDisA                            ⍝ create edit attributes
 Title ORIG
⍝
SCREEN:Cs←7,5,0⍴Ds←⍙C[1;18]                   ⍝ Set message line to no-blink
 Cs←2,5,0⍴Ds←MSG0
 Cs←4,1,0⍴Ds←DIST                             ⍝ column labels
 Cs←4,3,0⍴Ds←DISB                             ⍝ column formats
 Cs←4,2,0⍴Ds←DISL,0⍴Cs←7,2,0,⍴Ds←⍙C[1;16]     ⍝ row labels
 Cs←7,8,0⍴Ds←DISA                             ⍝ edit window attributes
 Cs←2,8,0⍴Ds←DISP                             ⍝ edit window
 ShowCell                                     ⍝ Highlight cell
⍝
⍝  Wait for and evaluate key strokes
⍝
Key:→((⍴⍙ktyp)<PRESS←⍙ktyp⍳⊂CHR←2↑Ds,0⍴Cs←3,1)⍴ANY
 →⍎PRESS⊃⍙key
⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
CF1:→REFRESH                                                ⍝ Rebuild screen
⍝
F1:→SCREEN,Help                                             ⍝ Help
⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
F8:→(27=1↑FindTxt←FindTxt Input 'Search for ...')⍴SCREEN          ⍝ Search
 Cs←2,5,0⍴Ds←'      Searching ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 (K L)←⍙AR ⍙AC
F81:→(⍙MC≥L←L+1)⍴F82
 →(⍙MR<K←K+L←1)⍴F83
F82:→(∨/FindTxt⍷⍎∊('' '⍕' 'FromDays')[1+⍙MP[3;L]],'⊃MAT[K;L]')↓F81
 (⍙AR ⍙AC)←K L
 ⍙CO←⍙CO⌈(⍙WC≥(+/⍙MP[1 2;⍙AC])-⍙MP[1;])⍳1
 →RESTART,⍙RO←⍙RO⌈⍙AR+⍙UR-⍙SR-4
F83:→SCREEN,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Not found: ',FindTxt
⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
CF9:Cs←2,6,0⍴Ds←⍕K←MISS∆N                                     ⍝ Chg MISS∆N
 →(27=1↑N←1 Input 'Enter new MISSING VALUE code')⍴SCREEN
 '→CF91' ⎕EA 'MISS∆N←⍎N'
 MAT[(K≡¨MAT)/⍳⍴MAT←,MAT]←MISS∆N
 MAT←(∆MR,∆MC)⍴MAT
 →REFRESH
CF91:→CF9,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'MISSING VALUE must be numeric'
⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
CF4:→Command↓SCREEN,REFRESH                                ⍝ APL command
⍝
F5:→(Menu 1)↓SCREEN,RESTART                                ⍝ Statistics menu
⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
F7:→Scr1,InsRow                                             ⍝ Insert row
SF7:→InsCol↓SCREEN,Scr1                                     ⍝ Insert column
⍝
F9:→(CopyMove 1)↓SCREEN,Scr1                                ⍝ Copy area
SF9:→(CopyMove 0)↓SCREEN,Scr1                               ⍝ Move area
⍝
F10:⍎((⍳0)≡K←(∧/⍙MA≥8)/⍳⍙MR)/'→F101,K←,⍙AR'                 ⍝ Delete rows
 →(GetYN 'Delete marked rows')↓SCREEN
 →F102,⍴⍙UL[1]←' '
F101:→(GetYN 'Delete current row')↓SCREEN
```

```
[ 81] F102:⍎(1+⍙MR=⍴K)⊃'→Scr0,DelRow K' '→SCREEN'
[ 82] ⍝
[ 83] SF10:⍎((⍳0)≡K←(∧⌿⍙MA≥8)/⍳⍙MC)/'→SF101,K←,⍙AC'           ⍝ Delete columns
[ 84]  →(GetYN 'Delete marked columns')↓SCREEN
[ 85]  →SF102,⍴⍙UL[1]←' '
[ 86] SF101:→(GetYN 'Delete current column')↓SCREEN
[ 87] SF102:⍎(1+⍙MC=⍴K)⊃'→Scr0,DelCol K' '→SCREEN'
[ 88] ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
[ 89] SHi:→(High 1)↓SCREEN,Scr2                                ⍝ Highl. field
[ 90] SHB:→(High 2)↓SCREEN,Scr2                                ⍝ Highl. block
[ 91] SHR:→(High 3)↓SCREEN,Scr2                                ⍝ Highl. row
[ 92] SHC:→(High 4)↓SCREEN,Scr2                                ⍝ Highl. column
[ 93] SHU:→(High 5)↓SCREEN,Scr2                                ⍝ Un-highlight
[ 94] SHS:→Shadow↓RESTART,REFRESH                              ⍝ Shadow
[ 95] ⍝
[ 96] CF:→Scr2,Mark 1                                          ⍝ Mark field
[ 97] CB:→Scr2,Mark 2                                          ⍝ Mark block
[ 98] CR:→Scr2,Mark 3                                          ⍝ Mark row
[ 99] CC:→Scr2,Mark 4                                          ⍝ Mark column
[100] CU:→Scr2,Mark 5                                          ⍝ Unmark
[101] CF8:→Rotate↓RESTART,REFRESH                              ⍝ Rotate
[102] ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
[103] F2:→SCREEN,CFLAG←0×SaveMat ORIG                            ⍝ Save
[104] SF2:→SCREEN,CFLAG←0×SaveMat ''                             ⍝ Save as ...
[105] F3:→Quit↓SCREEN,RC←0                                       ⍝ Quit
[106] F4:→(RC←SaveMat ORIG)↓SCREEN,0                             ⍝ File
[107] ⍝
[108] CF2:→(Menu 2)↓Scr1,RESTART                                 ⍝ File menu
[109] ⍝
[110] CF10:→Scr1,Menu 3                                          ⍝ Print menu
[111] ⍝
[112] CF3:→GetBlock↓SCREEN,REFRESH                               ⍝ Get Block
[113] ⍝
[114] SF3:→(27=1↑N←GetName '')⍴SCREEN                            ⍝ Put block
[115]  →(''≡K←CompBlock)⍴SF32
[116]  '→SF31' ⎕EA '⍎N,''←CompBlock'''
[117]  →SCREEN
[118] SF31:→SF3,0⍴Cs←11,1,0,⍴Cs←2,5,0⍴Ds←'Unable to save as ',N
[119] SF32:→SCREEN,0⍴Cs←11,1,0,⍴Cs←2,5,0⍴Ds←'Nothing to save'
[120] ⍝
[121] SF4:→(NewMat '')↓SCREEN,NEW,NEW              ⍝ New matrix
[122] ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
[123] ANY:→(0 GetEntry ⎕AF 1↑CHR)↓SCREEN,RESTART           ⍝ ENTER / ANY other
[124] Ret:→((⍙AR>⍙MR)∨⍙AC>⍙MC)⍴∆∆R1                        ⍝   key pressed
[125]  N←⊃MAT[⍙AR;⍙AC]
[126]  ⍎(⍙MP[3;⍙AC]=2)/'N←,FromDays N'
[127]  →(0 GetEntry⍕N)↓SCREEN,RESTART
[128] ∆∆R1:→(0 GetEntry '')↓SCREEN,RESTART
[129] ⍝
[130] F6:→Scr1,Bulk                                        ⍝ Bulk entry
[131] ⍝
[132] SF8:→Recode↓SCREEN,RESTART                           ⍝ Recode
[133] ⍝
[134] CF7:→(ToggType ⍙AC)↓SCREEN,RESTART                   ⍝ Change column type
[135] ⍝
```

```
[136] SF1:→Sort↓SCREEN,Scr1                                  ⍝ Sort
[137] ⍝
[138] SF5:→RESTART,MakeLabRow                                ⍝ Toggle col labels
[139] SF6:→RESTART,MakeLabCol                                ⍝ Toggle row labels
[140] CF5:→RESTART,EdLab                                     ⍝ Edit col labels
[141] ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
[142] CurLe:→GoHor,⍙CO←⍙CO⌊⍙AC←1⌈⍙AC-1                                ⍝ Scrolling
[143] CurRi:→GoHor,⍙CO←⍙CO⌈(⍙WC≥(+/⍙MP[1 2;⍙AC←1+⍙MC⌊⍙AC])-⍙MP[1;])⍳1
[144] CtlLe:→GoHor,(⍙AC ⍙CO)←1⌈¯1+⍙AC ⍙CO
[145] CtlRi:→GoHor,(⍙AC ⍙CO)←(1+⍙MC)⌊1+⍙AC ⍙CO
[146] Tab:→GoHor,⍙CO←⍙AC←⍙MC⌊+/(⍙WC+⍙MP[1;⍙CO])≥⍙MP[1;]
[147] STab:→GoHor,⍙CO←⍙AC←1⌈+/(⍙MP[1;⍙CO]-⍙WC)>⍙MP[1;]
[148] Home:→GoHor,⍙AC←⍙CO←1
[149] End:⍙CO←(⍙CO⌈((1+⍙SC-⍙UC)≥(+/⍙MP[1 2;⍙AC←⍙MC+1])-⍙MP[1;])⍳1)
[150] GoHor:⍎(⍙OC≠⍙CO)/'→Scr1,DISB←HorShift'
[151]  →SCREEN
[152] ⍝
[153] CurUp:→GoVer,⍙RO←⍙RO⌊⍙AR←1⌈⍙AR-1
[154] CurDn:→GoVer,⍙RO←⍙RO⌈1-⍙WR-⍙AR←1+⍙MR⌊⍙AR
[155] PgUp:→GoVer,(⍙AR ⍙RO)←1⌈1-⍙WR-⍙AR ⍙RO
[156] PgDn:⍙AR←(1+⍙MR)⌊⍙AR+⍙WR-1
[157]  →GoVer,⍙RO←1⌈(⍙RO+⍙WR-1)⌊⍙MR-⍙WR-2
[158] CtlUp:→GoVer,(⍙AR ⍙RO)←1⌈¯1+⍙AR ⍙RO
[159] CtlDn:→GoVer,(⍙AR ⍙RO)←1+⍙MR⌊⍙AR ⍙RO
[160] CtlHome:→GoVer,⍙AR←⍙RO←1
[161] CtlEnd:⍙RO←⍙RO⌈1-⍙WR-⍙AR←⍙MR+1
[162] GoVer:⍎(⍙OR≠⍙RO)/'→Scr1,DISL←VerShift'
[163]  →SCREEN
[164] ⍝
[165] CL:→Locate↓SCREEN,Scr1                                      ⍝ Goto field
[166] ⍝
[167] AltRi:→⍎(1+2=⎕NC 'DFLAG')⊃'Key' '0,RC←1'                     ⍝ Alt-cursor
[168] AltLe:→⍎(1+2=⎕NC 'DFLAG')⊃'Key' '0,RC←¯1'                    ⍝   valid for
[169] AltUp:→⍎(1+2=⎕NC 'DFLAG')⊃'Key' '0,RC←¯2'                    ⍝   draftsmen
[170] AltDo:→⍎(1+2=⎕NC 'DFLAG')⊃'Key' '0,RC←2'                     ⍝   only
[171] ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
[172] BSP: ⍝ N←UWE                      ⍝ Intentional error for debugging
[173] ⍝
[174] CF6:                              ⍝ Currently unused, but defined label
[175]  →Key
```

```
[  0] M←R AddLabel C;Mmat;Matt
[  1] ⍝
[  2] ⍝  Returns (mat attr), a submatrices of MAT and ⍙MA consisting of
[  3] ⍝  rows R and columns C, with row/column labels appended
[  4] ⍝
[  5]  (Mmat Matt)←(MAT[R;C])(8|⍙MA[R;C])
[  6]  ⍎(⍙UC>6)/'(Mmat Matt)←(⍙LC[R],Mmat)(0,Matt)'
[  7]  ⍎(⍙UR=3)/'(Mmat Matt)←((((⍙UC>6)/⍙LS),⍙LR[C]),[1]Mmat)(0,[1]Matt)'
[  8]  M←Mmat Matt
```

```
X Ass Y
⍝
⍝  Emulate X←¨Y
⍝
 MAT[X[1];X[2]]←⊂Y
```

```
RC←P BackLabel Mmat;R;C;SLR
⍝
⍝  Write submatrix Mmat back into MAT[R;C], P=R,C; RC: always 1
⍝  First row/column may belong to label vectors
⍝
 Cs←2,5,0ρDs←'     Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
 SLR←⊂''
 (R C)←P
 ⍎(⍙UR=3)/'(Mmat SLR)←(1↓[1]Mmat)(Mmat[1;])'
 →(⍙UC=6)ρ∆2
 (SLR ⍙LS)←(1↓SLR)(1↑SLR)
 ⍙LC[R]←Mmat[;1]
 Mmat←1↓[2]Mmat
∆2:⍎(⍙UR=3)/'⍙LR[C]←SLR'
 MAT[R;C]←Mmat
 RC←CFLAG←1
```

```
RC←Bulk;S
⍝
⍝  Bulk entry mode. Enter new values into successive fields,
⍝  creating new rows/columns if necessary. RC dummy
⍝
 S←0
 Cs←11,1,0ρCs←2,5,0ρDs←'Row- or columnwise (R/C) ?'
 →(∧/67 99 82 114≠K←1↑Ds,0ρCs←3,1)/⎕LC           ⍝ get answer
 →(67 99 82 114=K)/∆21,∆21,∆11,∆11               ⍝ cols=∆11, rows=∆21
⍝
⍝  Rowwise
⍝
∆11:RC←⍙CO←⍙AC←InsRow                   ⍝ Insert empty row, go to column 1
∆12:Cs←4,3,0ρDs←DISB←HorShift           ⍝ Adjust display
∆13:Cs←7,8,0ρDs←DISA←MakeDisA
 Cs←2,8,0ρDs←DISP←MakeDisP
 ShowCell
 →(1 GetEntry '')↓∆14                                      ⍝ Get new value
 ⍎(⍙MC<⍙AC←⍙AC+1)/'→∆11,⍙RO←⍙RO⌈(⍙UR-⍙SR-4)+⍙AR←⍙AR+1'        ⍝ New line
 →(⍙OC=⍙CO←⍙CO⌈((1+⍙SC-⍙UC)≥⍙MP[1;⍙AC+1]-⍙MP[1;])⍳1)↓∆12,∆13   ⍝ Next cell
∆14:⍎(⍙AC=1)/'⎕WA←DelRow ⍙AR'                  ⍝ Exit, delete row if empty
 →∆30
⍝
⍝  Columnwise
⍝
∆20:Cs←2,8,0ρDs←DISP←MakeDisP
∆21:→(RC←InsCol)↓∆25                       ⍝ Insert empty col, go to row 1
 ⍙RO←⍙AR←S←1
∆22:Cs←4,2,0ρDs←DISL←VerShift              ⍝ Adjust display
∆23:Cs←7,8,0ρDs←DISA←MakeDisA
```

```
 Cs←2,8,0⍴Ds←DISP←MakeDisP
 ShowCell
 →(1 GetEntry '')↓∆24                        ⍝ Get new value
 ⍎(⍙MR<⍙AR←⍙AR+1)/'→∆20,⍙AC←⍙AC+1'           ⍝ New column
 →(⍙OR=⍙RO←⍙RO⌈1-⍙WR-⍙AR)↓∆22,∆23             ⍝ Next cell
∆24:⍎(⍙AR=1)/'⎕WA←DelCol ⍙AC'                ⍝ Exit, delete column if empty
 →∆30
∆25:→S↓0
 (⍙AR ⍙AC)←(⍙AR ⍙AC)-1
⍝
∆30:Refresh                                  ⍝ Update the matrix
 DISB←HorShift
```

```
U←D CHISQ X;S;U;C1;C2;C3;N;V
⍝  Original program by Prof. H. Larson; modified for APL2
⍝  12/3/86 Accurate and fast evaluation of the cdf for the Chi Square cdf,
⍝  at vector X, D df.  For D = 1 or 2, the normal (GAUS) and exponential
⍝  distributions are used, respectively.  For D>2 and odd, the Lau sum,
⍝  Applied Statistics, v29, 1980, p113, algorithm 147, is used, which ap-
⍝  pears to be the same as Abramowitz and Stegun, p941, 26.4.6. For D>2 and
⍝  even, Abramowitz ∧ Stegun, p941, 26.4.5, is used selectively, depending
⍝  on D and ⌈/X; for certain combinations the Lau sum is again used.
⍝
 →(D=1 2)/∆1,∆2
 S←(X-D)÷(2×D)*0.5
 U←(⍴,X)⍴1
 C1←(⌈6.9×D*0.44)
 C2←((⌈/S)≤0)×((D≤70),D>70)/5 7
 C3←((⌈/S)>0)×⌊((D≤200),D>200)/2.5 3.1×⍟D
 →(2|D)/Odd
 →(((N←C1+⌈(⌈/S)×(C3+C2))≤¯1+D÷2)∨(D≥208))/Even
 →0,⍴U←1-(*-X÷2)×1++/((,X)∘.*⍳(D-2)÷2)÷×\((⍴,X)⍴0)∘.+2×⍳(D-2)÷2
Odd:N←C1+⌈(⌈/S)×(C2+C3)
 V←(*-X÷2)×+/×\1,(X÷2)∘.÷(D÷2)+⍳N⌈20
 →0,⍴U←V×((2×X÷○1)*0.5)××/X∘.÷(1+2×⍳(D-1)÷2)
Even:V←(*-X÷2)×+/×\1,(X÷2)∘.÷(D÷2)+⍳N⌈20
 →0,⍴U←,V××/(X÷2)∘.÷⍳D÷2
∆1:→0,⍴U←¯1+2×GAUSS X*0.5
∆2:U←1-*-X÷2
```

```
CLEAN
⍝
⍝  Cleanup workspace
⍝
 ⎕←''
 ⎕←'  1  Erase all functions and variables which ARE part of UEdit'
 ⎕←'  2  Erase all functions and variables which are NOT part of UEdit'
 ⎕←''
 ⎕←'  Type anything else to abort'
 ⎕←''
 ⍞←'  Your choice:  '
 '→∆3' ⎕EA '→((1 2=⍎14↓⍞),1)/∆1,∆2,∆3'
⍝
```

```
[ 13] Δ1:⎕WA←⎕EX⊃⍙nl3                          ⍝ Erase all functions/variables
[ 14]  →0,⎕EX⊃⍙nl2                             ⍝ which are part of UEdit
[ 15] ⍝
[ 16] Δ2:⎕WA←⎕EX⊃(Dtb¨⊂[2]⎕NL 2)~⍙nl2          ⍝ Erase all functions/variables
[ 17]  →0,⎕EX⊃(Dtb¨⊂[2]⎕NL 3)~⍙nl3             ⍝ which are not part of UEdit
[ 18] ⍝
[ 19] Δ3:⍞←'  Aborted'


[  0] CSVprep I;ROW;C
[  1] ⍝
[  2] ⍝  Create CSV vector from matrix/label vector HMAT[I;], enclose
[  3] ⍝  elements containing commas IN ⍝, replace ¯ by -, and append
[  4] ⍝  vector as nested element to OUTMAT
[  5] ⍝
[  6]  Cs←2,5,0⍴Ds←'     Preparing line ',(⍕I),' of ',⍕⍙MR+⍙UR-2
[  7]  ROW[C]←'⍝',¨ROW[C←(∨/¨',' =ROW←HMAT[I;])/⍳1↓⍴HMAT],¨'⍝'
[  8]  ROW[('¯'=ROW)/⍳⍴ROW←1↓∊',',¨⍕¨ROW]←'-'
[  9]  OUTMAT←OUTMAT,⊂ROW


[  0] ChgType COL;I;C
[  1] ⍝
[  2] ⍝  Try to make column label numeric, otherwise change matrix column
[  3] ⍝  to character type.
[  4] ⍝
[  5]  '' ⎕EA '→0,⍙LR[COL]←⍎COL⊃⍙LR'
[  6]  ⎕WA←'C' ToggType COL
[  7]  NoNum←NoNum,COL


[  0] ClearUL
[  1] ⍝
[  2] ⍝  Display 'upper left screen corner'
[  3] ⍝
[  4]  Cs←1,10,0⍴Ds←1,1,(¯1+⍙UR,⍙UC),2,⍙C[1;16]
[  5]  Cs←4,10,0⍴Ds←((⍙UC+4)⍴' '),,⊃⍙LS
[  6]  Cs←1,10,0⍴Ds←1,1,1,3,2,⍙C[2;16]
[  7]  Cs←4,10,0⍴Ds←⍙UL


[  0] RC←Command;R;R1;C;CMD
[  1] ⍝
[  2] ⍝  Enter and execute user's APL commands;  RC: 1 - excuted, 0 - escape
[  3] ⍝
[  4]  CMD←''
[  5] Δ1:R←'[',(⍕⍙AR),';',(⍕⍙AC),']'
[  6]  CMD←CMD Input 'Enter APL commands ( α = ',ORIG,R,',  ω = tagged area )'
[  7]  →(27=1↑CMD)⍴RC←0
[  8] ⍝
[  9]  Cs←2,5,0⍴Ds←'     Executing ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 10]  CMD←(∨\CMD≠' ')/CMD                              ⍝ Drop leading blanks
[ 11]  CMD←('αMAT',R)Replace CMD                        ⍝ Replace α
[ 12]  →('ω'∊CMD)↓Δ2                                    ⍝ Check for ω
[ 13]  →(0=1↓⍴R←1+(⍴⍙MA)⊤¯1+(8≤,⍙MA)/⍳⍴,⍙MA)⍴ΔE        ⍝ area marked ?
[ 14]  R←⊂¨R1←⊂[1]R                                       ⍝ Prepare indices
[ 15]  CMD←CMD[1],('⍵(R⊃¨⊂MAT)')Replace 1↓CMD              ⍝ Replace ⍵ in 1↓CMD
[ 16]  →('⍵'≠CMD[1])⍴∆2                                   ⍝ -> Normal execution
[ 17] ⍝
[ 18] ⍝  Work around the missing 'selective assignment' feature
[ 19]  '→∆E' ⎕EA 'CMD←⍎(1++/∧\CMD≠''←'')↓CMD'       ⍝ These two lines are for
[ 20]  '→∆E' ⎕EA 'R1 Ass¨CMD'                       ⍝   the 'production version'
[ 21] ⍝  CMD←⍎(1++/∧\CMD≠'←')↓CMD                    ⍝ Use these two lines
[ 22] ⍝  R1 Ass¨CMD                                  ⍝   for debugging
[ 23]  →∆3
[ 24] ⍝  Normal execution
[ 25] ∆2:'→∆E' ⎕EA '⍎CMD'                            ⍝ Production version
[ 26] ⍝  ∆2:⍎CMD                                     ⍝ Debugging version
[ 27] ⍝
[ 28] ∆3:→⍙OC←⍙OR←↑0,CFLAG←RC←1
[ 29] ⍝
[ 30] ∆E:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'     Invalid command'
```

```
[  0] OMAT←CompBlock;OLR;R;C
[  1] ⍝
[  2] ⍝  Return marked block with column and row labels appended if they exist
[  3] ⍝
[  4]  ⍎((⍳0)≡R←(∨/⍙MA≥8)/⍳⍙MR)/'→0,OMAT←'''''             ⍝ marked rows
[  5]  C←(∨⌿⍙MA≥8)/⍳⍙MC                                    ⍝ marked columns
[  6]  (OMAT OLR)←MAT[R;C]⍙LR
[  7]  ⍎(⍙UC>6)/'(OMAT OLR C)←(⍙LC[R],OMAT)(⍙LS,OLR)(1,C+1)'  ⍝ add row labels
[  8]  ⍎(⍙UR=3)/'OMAT←OLR[C],[1]OMAT'                       ⍝ add column labels
```

```
[  0] OMAT←CompMat MAT;OLR
[  1] ⍝
[  2] ⍝  Return MAT with column and row labels appended if they exist
[  3] ⍝
[  4]  (OMAT OLR)←MAT ⍙LR
[  5]  ⍎(⍙UC>6)/'(OMAT OLR)←(⍙LC,OMAT) (⍙LS,OLR)'
[  6]  ⍎(⍙UR=3)/'OMAT←OLR,[1]OMAT'
```

```
[  0] RC←CopyMove FLAG;R;C;K;A
[  1] ⍝
[  2] ⍝  Copy or move columns/rows to cursor position
[  3] ⍝  FLAG: 1-copy, 0-move     RC: 1-ok, 0-error
[  4] ⍝
[  5]  →((⍳0)≡R←(∨/⍙MA≥8)/⍳1↑⍴⍙MA)⍴∆E2                      ⍝ rows with marks
[  6]  →(0=∧/,⍙MA[R;C←(∨⌿⍙MA≥8)/⍳1↓⍴⍙MA]≥8)⍴∆E1             ⍝ columns with marks
[  7]  Cs←2,5,0⍴Ds←'     Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[  8]  →(⍙MR ⍙MC=(⍴R),⍴C)/∆2,∆1
[  9] ⍝
[ 10] ∆E1:→∆E0,MSG←'Unable to copy/move rows and columns simultaneously'
[ 11] ∆E2:MSG←'No rows or columns highlighted'
[ 12] ∆E0:→RC←0,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG
[ 13] ⍝
[ 14] ∆1:K←((⍙AR-1)↑⍳⍙MR),R,(⍙AR-1)↓⍳⍙MR                    ⍝ Copy/move rows
[ 15]  A←R+(⍴R)×R≥⍙AR
```

```
 ⍎(~FLAG)/'K←K[(⍳⍴K)~A]'
 (MAT ⍙MA ⍙DM ⍙DA ⍙LC)←(MAT[K;])(⍙MA[K;])(⍙DM[K;])(⍙DA[K;])(⍙LC[K])
 →FLAG↓Δ11
 ⍙MA[A;]←8|⍙MA[A;]
 ⍙DA[A;]←(¯1↓2+⍙MP[2;])/⎕AF ⍙C[1;1+⍙MA[A;]]
Δ11:⍙MR←↑⍴K
 DISL←VerShift
 →0,RC←CFLAG←1
⍝
Δ2:K←((⍙AC-1)↑⍳⍙MC),C,(⍙AC-1)↓⍳⍙MC                    ⍝ Copy/move columns
 A←C+(⍴C)×C≥⍙AC
 ⍎(~FLAG)/'K←K[(⍳⍴K)~A]'
 (MAT ⍙MA ⍙LR)←(MAT[;K])(⍙MA[;K])(⍙LR[K])
 (⍙DM ⍙DA ⍙CL ⍙CT)←(⍙DM[;R])(⍙DA[;R])(⍙CL[;R])(⍙CT[R←∊GC¨K])
 ⍙MP←⍙MP[;K],0,3,0,0,0
 ⍙MP[1;]←0,¯1↓+\2+⍙MP[2;]
 →FLAG↓Δ21
 ⍙MA[;A]←8|⍙MA[;A]
 ⍙DA[;∊GC¨A]←(2+⍙MP[2;A])/⎕AF ⍙C[1;1+⍙MA[;A]]
Δ21:⍙MC←↑⍴K
 ⍙CL[RC←CFLAG←1;]←∊(2+¯1↓⍙MP[2;])↑¨(⍕¨⍳⍙MC)
 DISB←HorShift


RC←Cross;RCM;ORCM;COL;MSG;PAR;Ind1;Ind2;Uni1;Uni2;OUni1;OUni2;RI;I;RATT;
   ⍙SP;MM;CM
⍝
⍝  Crosstabulation of two matrix columns;  RC: 1- ok, 0 - escape
⍝
 COL←''
Δ1:→(27=1↑COL←COL Input 'Enter column numbers')⍴RC←0
 '→ΔE1' ⎕EA '→(2≠⍴COL←⍎COL)⍴ΔE2'            ⍝ must be numeric, 2 columns
 →(∧/COL∊⍳⍙MC)↓ΔE3                           ⍝ columns must exist
⍝
 PAR←askΔpar¨COL                             ⍝ Get column parameters
⍝
 I←''
Δ2:→(27=1↑I←I Input 'conditioning columns ... ? (ENTER if none)')⍴0
 ⍎(I≡,' ')/'→Δ3,⍴CM←MAT[;COL]'                    ⍝ empty: no condition
 →(27=1↑RI←GetCond I)⍴0                           ⍝ get condition vector
 →(¯1=1↑RI)⍴Δ2
 →(0=↑⍴CM←RI⌿MAT[;COL])⍴ΔE4                       ⍝ extract matching elts
⍝
Δ3:Cs←2,5,0⍴Ds←'     Crosstabulating ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 (Ind1 Uni1)←((COL[1])(⊃PAR[1]))MakeIndex CM[;1]     ⍝ get index/label
 (Ind2 Uni2)←((COL[2])(⊃PAR[2]))MakeIndex CM[;2]     ⍝   vectors
 RI←RI[⍋RI←Ind2+(Ind1-1)×⍴Uni2]
 RCM←((⍴Uni1),⍴Uni2)⍴RI freqΔcnt(⍴Uni1)×⍴Uni2        ⍝ RCM - cross matrix
⍝  Save values for pooling
 ORCM←RCM
 OUni1←Uni1
 OUni2←Uni2
⍝  Create and edit cross mat
 ⍎(MM←(GetTitle COL[1]),'_',GetTitle COL[2]),'←CtPrep RCM'
 ⎕WA←USER MM
```

```
[ 30]  →0,RC←1
[ 31] ⍝
[ 32] ⍝  Error handling
[ 33] ∆E1:→∆E0,MSG←'Input must be a numeric vector'
[ 34] ∆E2:→∆E0,MSG←'Specify 2 columns, please'
[ 35] ∆E3:MSG←'Non existing column number specified'
[ 36] ∆E0:→∆1,0ρCs←11,1,0ρCs←2,5,0ρDs←MSG
[ 37] ∆E4:→∆2,0ρCs←11,1,0ρCs←2,5,0ρDs←'Nothing left to count !?!?'
```

```
[  0] CM←CtPrep R;E;S;X;D;BL;RT;C;R1;CHI
[  1] ⍝
[  2] ⍝  Create cross matrix CM, attribute matrix RATT
[  3] ⍝
[  4]  E←((+/R)∘.×+⌿R)÷RT←+/+/R                     ⍝ E - expected frequencies
[  5]  S←(R-E)÷E*0.5                                ⍝ S - standardized residuals
[  6]  X←+/+/((R-E)*2)÷E                            ⍝ X - chi-square
[  7]  D←(¯1+ρUni1)×¯1+ρUni2                        ⍝ D - degrees of freedom
[  8]  →(D=0)ρ∆1,CHI←MISS∆N                         ⍝ If D>0
[  9]  CHI←1-D CHISQ X                              ⍝   CHI - p-level
[ 10] ⍝
[ 11] ⍝  Generate display matrices
[ 12] ⍝
[ 13] ∆1:BL←(' '),(1+¯1↑ρR)ρMISS∆N
[ 14]  C←1↓ρCM←(1,2+ρUni2)ρ(' '),(⍕¨Uni2),⊂'total'
[ 15]  R1←3×ρUni1
[ 16]  CM←CM,[1](R1,C)ρ(⍕¨Uni1),R,(+/R),(⍕¨Uni1),(S),((+/R)÷RT),((ρUni1),C)ρBL
[ 17]  RATT←(R1,C-1)ρ(((ρUni1),C-2)ρ0),6,(1++⌿1.96 2.54∘.≤|S),0,((ρUni1),C-1)ρ0
[ 18]  CM←CM,[1](⊂'total'),(+⌿R),RT
[ 19]  CM←CM,[1](⊂'column +'),((+⌿R)÷RT),MISS∆N
[ 20]  CM←CM,[1]BL
[ 21]  CM←CM,[1](⊂'d.o.f.'),D,(¯1↑ρR)ρMISS∆N
[ 22]  CM←CM,[1](⊂'Chi-sq'),X,(¯1↑ρR)ρMISS∆N
[ 23]  CM←CM,[1](⊂'signif'),CHI,(¯1↑ρR)ρMISS∆N
[ 24]  RATT←RATT,[1]6,[1](5,C-1)ρ0
[ 25]  ⍙SP←1,[0.5](¯1↑ρRATT)ρ2                          ⍝ 2 decimals by default
```

```
[  0] RC←DelCol C;N
[  1] ⍝
[  2] ⍝  Delete columns indexed by C;  RC dummy
[  3] ⍝
[  4]  Cs←2,5,0ρDs←'     Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[  5]  (⍙LR MAT ⍙MA)←(⍙LR[C])(MAT[;C])(⍙MA[;C←(⍳⍙MC)~C])
[  6]  (⍙DM ⍙DA)←(⍙DM[;N])(⍙DA[;N←∊GC¨C])
[  7]  (⍙CL ⍙CT ⍙AC)←(⍙CL[;N])(⍙CT[N])(⍙AC⌊⍙MC←↑ρ,C)
[  8]  ⍙CO←⍙CO⌊⍙MC
[  9]  ⍙MP←⍙MP[;C],0,3,0,0,0
[ 10]  ⍙MP[RC←CFLAG←1;]←0,¯1↓+\2+⍙MP[2;]
[ 11]  ⍙CL[1;]←∊(2+¯1↓⍙MP[2;])↑¨⍕¨⍳⍙MC
[ 12]  DISB←HorShift
[  0] RC←DelRow N;K
[  1] ⍝
[  2] ⍝  Delete rows indexed by N;  RC dummy
[  3] ⍝
[  4]  Cs←2,5,0ρDs←'      Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[  5]  (⍙LC ⍙MA ⍙DA ⍙DM MAT)←(⍙LC[N])(⍙MA[N;])(⍙DA[N;])(⍙DM[N;])(MAT[N←(⍳⍙MR)~N;])
[  6]  RC←CFLAG←↑1,⍙AR←⍙AR⌊⍙MR←↑ρ,N
[  7]  ⍙RO←⍙RO⌊⍙MR
[  8]  DISL←VerShift
```

```
[  0] W←DisCol;N
[  1] ⍝
[  2] ⍝  Convert matrix column to character type for display
[  3] ⍝  Update width entry in parameter matrix ⍙MP
[  4] ⍝  W       column width
[  5] ⍝  C,COL  column number, converted column (global)
[  6] ⍝
[  7]  →(∆1,∆2,∆3)[1+⍙MP[3;C]]
[  8] ∆1:→∆4,ρCOL←(¯2↑1,ρCOL)ρCOL←⊃MAT[;C]                 ⍝ Character column
[  9] ∆1:→∆4,ρCOL←(2↑(ρCOL),1)ρCOL←⊃MAT[;C]                ⍝ Character column
[ 10] ∆2:→∆4,ρCOL←NumToChar MAT[;C]                        ⍝ numeric column
[ 11] ∆3:COL←FromDays MAT[;C]                              ⍝ date column
[ 12] ∆4:W←2+⍙MP[2;C]←3⌈(↑ρC⊃⍙LR)⌈¯1↑ρCOL                  ⍝ column width
```

```
[  0] RC←Draft;RCM;ORCM;COL;MSG;PAR;IU;LI1;LI2;LU1;LU2;Ind1;Ind2;Uni1;OUni1;Uni2;
         OUni2;LR;RI;RATT;DFLAG;PAR;C1;C2;TOT;∆SP;MM
[  1] ⍝
[  2] ⍝  Crosstabulation of multiple matrix columns;  RC: 1 - ok,  0 - escape
[  3] ⍝
[  4]  COL←''
[  5] ∆1:→(27=1↑COL←COL Input 'Enter column numbers - ENTER for all')ρRC←0
[  6]  ⍎(COL≡,' ')/'COL←⍕⍳⍙MC'
[  7]  '→∆E1' ⎕EA '→(2≥ρCOL←⍎COL)ρ∆E2'              ⍝ must be numeric, > 2 columns
[  8]  →(∧/COL∊⍳¯1↑ρMAT)↓∆E3                         ⍝ columns must exist
[  9] ⍝
[ 10] ∆2:PAR←ask∆par¨COL                             ⍝ Get classific. parameters
[ 11] ⍝
[ 12]  Cs←2,5,0ρDs←'      Crosstabulating ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[ 13]  LR←(TOT,TOT←¯1+↑ρCOL)ρC1←0
[ 14]  IU←⊃(COL,¨⊂¨PAR)MakeIndex¨⊂[1]MAT[;COL]
[ 15]  (LI1 LI2)←(¯1↓IU[;1])(1↓IU[;1])
[ 16]  (LU1 LU2)←(⍉(TOT,TOT)ρ¯1↓IU[;2])((TOT,TOT)ρ1↓IU[;2])
[ 17] ∆3:→(TOT<C1←1+C2←C1)ρ∆5
[ 18]  (Ind1 Uni1)←⊃¨LI1[C1]LU1[C1;1]
[ 19] ∆4:→(TOT<C2←C2+1)ρ∆3
[ 20]  (Ind2 Uni2)←⊃¨LI2[C2]LU2[1;C2]
[ 21]  RI←RI[⍋RI←Ind2+(Ind1-1)×ρUni2]
[ 22]  →∆4,LR[C1;C2]←⊂((ρUni1),ρUni2)ρRI freq∆cnt(ρUni1)×ρUni2
[ 23] ⍝
[ 24] ⍝  Display tables, use Alt-Cursor for scrolling
[ 25] ∆5:C1←C2←DFLAG←RC←1
[ 26] ∆6:(ORCM OUni1 OUni2)←(RCM Uni1 Uni2)←⊃¨LR[C1;C2]LU1[C1;C2]LU2[C1;C2]
[ 27]  ⍎(MM←(GetTitle COL[C1]),'_',GetTitle COL[1+C2]),'←CtPrep RCM'
[ 28]  DFLAG←USER MM
[ 29]  LR[C1;C2]←⊂RCM
[ 30]  LU1[C1;C2]←⊂Uni1
[ 31]  LU2[C1;C2]←⊂Uni2
[ 32]  →(DFLAG=0 1 ¯1 2 ¯2)/0,∆Ri,∆Le,∆Do,∆Up
[ 33] ∆Ri:→∆6,C2←TOT⌊C2+1
[ 34] ∆Le:→∆6,C2←C1⌈C2-1
[ 35] ∆Do:→∆6,C1←C2⌊C1+1
[ 36] ∆Up:→∆6,C1←1⌈C1-1
[ 37] ⍝
[ 38] ⍝  Error handling
[ 39] ∆E1:→∆E0,MSG←'Input must be a numeric vector'
[ 40] ∆E2:→∆E0,MSG←'Specify more than 2 columns, please'
[ 41] ∆E3:MSG←'Non existing column number specified'
[ 42] ∆E0:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG


[  0] Z←Dtb X
[  1] ⍝
[  2] ⍝  Drop trailing blanks from an array X
[  3] ⍝
[  4]  Z←((-⍴⍴X)↑⌈/(1⌈⍴Z)↑Z←,1-(X=' ')⊥1)↓X


[  0] RC←EdLab;W;N
[  1] ⍝
[  2] ⍝  Edit column labels in bulk mode;  RC dummy
[  3] ⍝
[  4]  ⍙WR←⍙SR-3+⍙UR←2+⍙AC←⍙CO←RC←1
[  5] ∆1:DISB←HorShift
[  6]  Cs←7,8,0⍴Ds←(DISA←MakeDisA),0⍴Cs←1,8,0⍴Ds←⍙UR,⍙UC,⍙WR,⍙WC,2,⍙C[1;1]
[  7]  Cs←2,8,0⍴Ds←DISP←MakeDisP
[  8]  N←⍙UC+⍙MP[1;⍙AC]-⍙MP[1;⍙CO]
[  9]  Cs←1,1,0⍴Ds←1,N,2,(1+⍙SC-N)⌊⍙MP[2;⍙AC],2,⍙C[1;16]
[ 10]  Cs←7,1,0⍴Ds←⍙C[2;16]
[ 11] ⍝
[ 12]  →(27=1↑N←(⍙AC⊃⍙LR)Input 'Enter new label for column ',⍕⍙AC)⍴∆2
[ 13]  ⍙LR[⍙AC]←⊂N
[ 14]  ⍙CL[2;⍙MP[1;⍙AC]+⍳W]←(W←PlaceEntry ⍙AC)↑N
[ 15]  ⍎(⍙MC≥⍙AC←⍙AC+1)/'→∆1,⍙CO←⍙CO⌈(⍙WC≥(+/⍙MP[1 2;⍙AC])-⍙MP[1;])⍳1'
[ 16]  ⍙AC←⍙AC-1
[ 17] ∆2:⍎(⍙AC≠1)/'CFLAG←1'


[  0] RC←FreqTab;CLR;COL;I;UNI;R;MSG;ELab;PAR;CUM;REL;⍙SP;RATT;FM;MM;RI
[  1] ⍝
[  2] ⍝  Frequency counts on a specified matrix column
[  3] ⍝  RC:  1 - ok,  0 - escape
[  4] ⍝
[  5]  →(27=1↑COL←ask∆col 'Enter column number')⍴RC←0
[  6]  →(COL≡,' ')⍴0
[  7] ⍝
[  8]  →(27=1↑PAR←ask∆par COL)⍴0                        ⍝ Get classific. parameters
[  9] ⍝
[ 10] ⍝  Conditional freq tabs ?
[ 11]  I←''
[ 12] ∆1:→(27=1↑I←I Input 'conditioning columns ... ? (ENTER if none)')ρ0
[ 13]  ⍎(I≡,' ')/'→∆2,0ρI←MAT[;COL]'                ⍝ Empty: no condition
[ 14]  →(27=1↑RI←GetCond I)ρ0                        ⍝ Get condition
[ 15]  →(¯1=1↑RI)ρ∆1
[ 16]  →(0=ρI←RI/MAT[;COL])ρ∆E1                      ⍝ Extract matching elements
[ 17] ⍝
[ 18] ⍝  Now I is the vector to perform the frequency count on
[ 19] ⍝  PAR holds the classification parameters if I is numeric
[ 20] ⍝
[ 21] ∆2:Cs←2,5,0ρDs←'     Counting ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[ 22]  (R UNI)←(COL PAR)MakeIndex I             ⍝ create index vector and labels
[ 23]  CUM←+\R←R[⍋R]freq∆cntρUNI                ⍝ cumulative/distinct frequencies
[ 24]  REL←R÷ρI                                 ⍝ relative frequencies
[ 25] ⍝
[ 26] ⍝  Prepare matrix display FM, and call USER recursive
[ 27]  FM←1 5ρ(' ')('Freq')('Rel.')('Cum.')(' ')
[ 28]  FM←FM,[1]⍉(5,ρR)ρUNI,R,REL,CUM,((R×40≥⌈/R)+⌈R×(40<⌈/R)×40÷⌈/R)ρ¨'█'
[ 29]  FM←FM,[1](' '),(3ρMISS∆N),' '
[ 30]  FM←FM,[1](⊂'total'),(+/R),(2ρMISS∆N),' '
[ 31]  RATT←(¯1+ρFM)ρ0
[ 32]  ⍙SP←2 4ρ1,1,1,0,99,2,99,99
[ 33]  ⍎(MM←'FR_',GetTitle COL),'←FM'
[ 34]  ⎕WA←USER MM
[ 35]  →0,RC←1
[ 36] ⍝
[ 37] ⍝  Error handling
[ 38] ∆E1:→∆1,0ρCs←11,1,0ρCs←2,5,0ρDs←'Nothing left to count !?!?'
```

```
[  0] YMD←FromDays DAYS;D;M;Y;D1;Y1;I;W
[  1] ⍝
[  2] ⍝  Convert days since 02/29/0000 to date array 'MM-DD-YYYY' or similar
[  3] ⍝  format according to DATE; sets MISS∆N to ' '
[  4] ⍝
[  5]  DAYS[W←(MISS∆N=DAYS)/⍳ρDAYS←,⌊DAYS]←1
[  6]  D1←DAYS-(365×Y)+-/⌊(Y←⌊DAYS÷365.2425)∘.÷4 100 400
[  7]  →(×ρI←(D1≥366)/⍳ρD1)↓∆1
[  8]  D1[I]←DAYS[I]-(365×Y1)+-/⌊(Y1←Y[I]←Y[I]+1)∘.÷4 100 400
[  9] ∆1:→(×ρI←(D1≤0)/⍳ρD1)↓∆2
[ 10]  D1[I]←DAYS[I]-(365×Y1)+-/⌊(Y1←Y[I]←Y[I]-1)∘.÷4 100 400
[ 11] ∆2:M←(31 30 31 30 31 31 30 31 30 31 31 29/2⌽⍳12)[D1]
[ 12]  D←D1-(306 337 0 31 61 92 122 153 184 214 245 275)[M]
[ 13]  YMD←(¯1↓∊((3,1,1)[DATE]ρ¨'0'),¨⊂'6-')⍕((Y+M≤2),M,[1.5]D)[;DATE]
[ 14]  YMD[W;]←' '
```

```
[  0] Z←GAUSS X;A;B;C;D
[  1] ⍝  Orig. program by Prof. H. Larson; modified for APL2
[  2] ⍝  Evaluates normal cdf at vector X. For |X<4, 26.2.11 in Abramowitz and
[  3] ⍝  Stegun, p. 932, is used. For large X, the continued fraction in Wall,
[  4] ⍝  p. 357, 92.11, is used at depth 16.  Appears to give at least 13 signi-
[  5] ⍝  ficant figures.
[  6] ⍝
[  7]  →((ρ,A←(|X<4)/Z←X←,X)=ρ,X)/1+⎕LC
[  8]   B←(|X≥4)/X
[  9]   →((⍴,B)=⍴X)/BIG
[ 10]   A←0.5+(÷((*A*2)×○2)*0.5)×+/((A∘.*¯1+2×⍳C+1)÷((⍴A),C+1)⍴×\¯1+2×⍳(C←10
          ⌈⌈10×⌈/|A)+1)
[ 11]   Z[(|X<4)/⍳⍴X]←A
[ 12]   →((⍴,A)=⍴X)/0
[ 13]  BIG:C←16589790 56295540 52050600 19934640 3680160 341952 15232 256
[ 14]   D←2027025 32432400 75675600 60540480 21621600 3843840 349440 15360 256
[ 15]   B←1-(B÷2×((○2)××B*2)*0.5)×(+/((0.5×,B*2)∘.*0,⍳7)×((⍴,B),8)⍴C)÷+/((0.5×,
          B*2)∘.*0,⍳8)×((⍴,B),9)⍴D
[ 16]   Z[(|X≥4)/⍳⍴X]←B
```

```
[  0] R←GC C
[  1] ⍝
[  2] ⍝  Return a vector of display column numbers corresponding
[  3] ⍝  to matrix column C; called always with ¨
[  4] ⍝
[  5]  R←⍙MP[1;C]+⍳2+⍙MP[2;C]
```

```
[  0] RC←GSout;FILE;COL
[  1] ⍝
[  2] ⍝  Export matrix column to an APL2 vector (numeric column) or a
[  3] ⍝  character matrix (character column) for use with GRAFSTAT
[  4] ⍝  RC: always 0
[  5] ⍝
[  6]  →(27=1↑COL←ask∆col 'Enter column number to export')⍴RC←0
[  7]  →(COL≡,' ')⍴0
[  8] ⍝
[  9]  FILE←''
[ 10] ∆1:→(27=1↑FILE←FILE Input 'Enter the name of the vector/matrix')⍴0
[ 11]  →(0≠⎕NC FILE)⍴∆Er
[ 12]  ⍎FILE,'←⊃MAT[;COL]'
[ 13]  →0
[ 14] ⍝  Error handling
[ 15] ∆Er:→∆1,0⍴Cs←11,1,0,⍴Cs←2,5,0⍴Ds←'Name is already in use'
```

```
[  0] RC←GetBlock;R;C;N;NMAT;K
[  1] ⍝
[  2] ⍝  Insert a matrix into the current row or column location
[  3] ⍝  RC: 1 - ok,  0 - escape
[  4] ⍝
[  5] ∆1:→(27=1↑N←'' Input 'Enter name of the APL matrix to insert')⍴RC←0
[  6]  ⍎(2≠⎕NC N)/'→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←''No matrix '',N,'' found'''
[  7]  NMAT←⍕¨⍎N
[  8] ⍝
[  9]  Cs←11,1,0⍴Cs←2,5,0⍴Ds←'insert Rows, Columns, Overlay (R/C/O) ?'
[ 10]  →(∧/67 79 82 99 111 114≠K←1↑Ds,0⍴Cs←3,1)/⎕LC
[ 11]  Cs←2,5,0⍴Ds←'     Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 12]  RC←CFLAG←1
[ 13]  →(82 114 67 99 79 111=K)/∆2,∆2,∆6,∆6,∆10,∆10
[ 14] ⍝
[ 15] ⍝  Insert rows, a vector is considered a 1×(⍴V) matrix
[ 16] ∆2:(R C)←⍴NMAT←(¯2↑1,⍴NMAT)⍴NMAT
[ 17]  ⍙AC←⍙CO←1
[ 18]  ⍎(⍙MC<C)/'NMAT←(C←⍙MC)↑[2]NMAT'
[ 19]  TryNum¨(0≠C↑⍙MP[3;])/⍳C
[ 20]  ⍎(⍙MC>C)/'NMAT←NMAT,[2](R,⍙MC-C)⍴(MISS∆N,'' '')[2-⍙MP[3;(⍳⍙MC)~⍳C]]'
[ 21] ∆3:MAT←((⍙AR-1)↑[1]MAT),[1]NMAT,[1](⍙AR-1)↓[1]MAT
[ 22]  ⍙MA←((⍙AR-1)↑[1]⍙MA),[1]((R,⍙MC)⍴0),[1](⍙AR-1)↓[1]⍙MA
[ 23]  →(⍙UC=6)⍴0
[ 24]  ⍙LC←((⍙AR-1)↑⍙LC),(R⍴' '),(⍙AR-1)↓⍙LC
[ 25]  →0
[ 26] ⍝
[ 27] ⍝  Insert columns, a vector is considered a (⍴V)×1 matrix
[ 28] ∆6:(R C)←⍴NMAT←(2↑(⍴NMAT),1)⍴NMAT
[ 29]  ⍎(⍙MR<R)/'→∆7,⍴NMAT←⍙MR↑[1]NMAT'
[ 30]  ⍎(⍙MR>R)/'NMAT←NMAT,[1]((⍙MR-R),C)⍴(MISS∆N,'' '')[2-Num¨NMAT[R;]]'
[ 31] ∆7:MAT←(X(⍙AC-1)↑[2]MAT),[2]NMAT,[2](⍙AC-1)↓[2]MAT
[ 32]  ⍙MA←((⍙AC-1)↑[2]⍙MA),[2]((⍙MR,C)⍴0),[2](⍙AC-1)↓[2]⍙MA
[ 33]  ⍙MP←((⍙AC-1)↑[2]⍙MP),[2](⍉(C,5)⍴0,0,0,'C',99),[2](⍙AC-1)↓[2]⍙MP
[ 34]  →(⍙UR=2)⍴0
[ 35]  ⍙LR←((⍙AC-1)↑⍙LR),(C⍴' '),(⍙AC-1)↓⍙LR
[ 36]  →0
[ 37] ⍝
[ 38] ⍝  Overlay block, cursor in upper left corner, vector = 1×⍴V matrix
[ 39] ∆10:(R C)←⍴NMAT←(¯2↑1,⍴NMAT)⍴NMAT
[ 40]  ⍎(R>K←1+⍙MR-⍙AR)/'NMAT←K↑[1]NMAT'
[ 41]  ⍎(C>K←1+⍙MC-⍙AC)/'C←1↓⍴NMAT←K↑[2]NMAT'
[ 42]  TryNum¨((⍙AC-1)↓0≠K↑⍙MP[3;])/(⍙AC-1)↓⍳K←⍙AC+C-1
[ 43]  MAT[¯1+⍙AR+⍳↑⍴NMAT;¯1+⍙AC+⍳1↓⍴NMAT]←NMAT
```

```
[  0] Col←GetColor
[  1] ⍝
[  2] ⍝  Get color attribute for row or column, Col ← 0 if escape
[  3] ⍝
[  4] ∆1:Col←'' Input 'Enter color level ( 0 ... 6 )'
[  5]  ⍎(27=1↑Col)/'→Col←0'
[  6]  '→∆E' ⎕EA '→(0≠⍴⍴Col←⍎Col)⍴∆E'
[  7]  →(Col∊0,⍳6)⍴0
[  8] ⍝  Error handling
[  9] ∆E:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Input must be ∊ [0,6]'
```

```
[  0] R←GetCond I;MSG;K;L;N;V
[  1] ⍝
[  2] ⍝  Get conditioning vector for FreqTabs, CrossTabs, i.e.,
[  3] ⍝  create a boolean 'yes/no' of rows to include
[  4] ⍝  R  - 27 = Escape pressed → do nothing
[  5] ⍝        ¯1 = Syntax error in I
[  6] ⍝        else boolean vector indicating matrix rows to choose
[  7] ⍝
[  8] ⍝  Parse conditioning vector
[  9]  →(∧/I∊'0123456789()/~∨∧⍱⍲, ')↓∆E1         ⍝ all symbols valid ?
[ 10]  N←(K←L←I)∊'0123456789'                    ⍝ extract digit positions
[ 11]  K[(~N)/⍳⍴K]←' '                           ⍝ cancel operators
[ 12]  K←,⍎K                                     ⍝ vector of numbers
```

```
 →(∧/K∊⍳⍙MC)↓∆E2                                ⍝ valid column numbers ?
 L[N/⍳⍴L]←'1'                                   ⍝ all digits ← '1'
 L←(N⍲0,¯1↓N)/L                                 ⍝ reduce multiple digits
 '→∆E1' ⎕EA '→(∧/0 1≠⍎L)⍴∆E1'                   ⍝ valid relation ?
⍝
 (I R)←1 27
∆1:→(R=1↑V←GetCrit K[I])⍴0                      ⍝ Get criteria for each vector
 L[L⍳'1']←⊂'(',(⍕V),')'                         ⍝ Put it into template
 →((⍴K)≥I←I+1)⍴∆1
 →0,R←∊⍎∊L                                      ⍝ Execute template
⍝
∆E1:→∆E0,MSG←'Syntax error'
∆E2:MSG←'Invalid column numbers specified'
∆E0:Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG
 R←¯1


R←GetCrit I;C;V;NOT;MSG
⍝
⍝  Get conditioning criteria for column I
⍝  R - 27 = Escape pressed, cancel operation
⍝      else  boolean vector indicating matrix rows to choose
⍝
 V←''
∆1:→((R←27)=1↑V←V Input 'Enter conditional criteria for column ',⍕I)⍴0
 ⍎(V≡,' ')/'→0,R←⍙MR⍴1'                ⍝ empty vector: unconditioned
 V[(V='¨')/⍳⍴V]←'⍝'                     ⍝ Replace ¨ by ⍝, parse vector, delete
 V←Dtb¨(Parse V),¨' '                   ⍝   trailing blanks of each element
⍝
 R←⍙MR⍴NOT←0
 ⍎(V[1]≡⊂,'~')/'→(0=⍴V←(NOT←1)↓V)⍴0'              ⍝ Check for NOT condition
⍝
 C←MAT[;I]
 →(1 2=⍙MP[3;I])/∆4,∆3
⍝
 →0,R←|NOT-C∊V                                    ⍝ Character column
⍝
∆3:→(∨/0=V←∊ToDays¨V)↓∆5,∆E3                      ⍝ Date column
⍝
∆4:⍎((,' ')≡∊V)/'→∆5,V←2⍴MISS∆N'                  ⍝ Numeric column
 '→∆E1' ⎕EA 'V←⍎¨V'
∆5:→(2|⍴V)⍴∆E2                                    ⍝ Argument pairs only
 V←((0.5×⍴V),2)⍴V
 →0,R←|NOT-∨⌿(V[;1]∘.≤C)∧V[;2]∘.≥C
⍝
∆E1:→∆E0,MSG←'Limits must be numerical'
∆E2:→∆E0,MSG←'Invalid number of arguments'
∆E3:MSG←'Invalid date specified'
∆E0:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG
```

```
RC←T GetEntry N;K;L;R
⍝
⍝  Enter new value for active cell
⍝  T: 0 - normal entry         RC: 0 - escape
⍝     1 - bulk mode entry          1 - ok
⍝
Δ1:→(27=1↑N←N Input 'Enter new value for active cell')ρRC←0
⍝
 N←('αMAT[',(⍕⍙AR),';',(⍕⍙AC),']')Replace N              ⍝ replace α
 →('ω'∊N)↓Δ2                                             ⍝ replace ω
 →(0=1↓ρR←1+(ρ⍙MA)⊤¯1+(8≤,⍙MA)/⍳ρ,⍙MA)ρΔ2
 N←('ω((⊂¨⊂[1]R)⊃¨⊂MAT)')Replace N
⍝
Δ2:⍎(⍙AR>⍙MR)/'⎕WA←InsRow'                               ⍝ add new row ?
 ⍎(⍙AC>⍙MC)/'⎕WA←InsCol ''C'''                           ⍝ add new column ?
 →(Δ3,Δ4,Δ5)[1+⍙MP[3;⍙AC]]
⍝
Δ3:→Δ6,MAT[⍙AR;⍙AC]←⊂N                                   ⍝ character input
⍝
Δ4:⍎(N≡,' ')/'→Δ6,MAT[⍙AR;⍙AC]←MISSΔN'                   ⍝ numeric input
 '→ΔE' ⎕EA '→Δ6,MAT[⍙AR;⍙AC]←⍎N'
⍝
Δ5:⍎(N≡,' ')/'→Δ6,MAT[⍙AR;⍙AC]←MISSΔN'                   ⍝ date input
 →(0=N←ToDays N)ρΔE
 MAT[⍙AR;⍙AC]←N
⍝
Δ6:→T↓Δ7                                                        ⍝ truncate
 →Δ8,⍙DM[⍙AR;⍙MP[1;⍙AC]+⍳⍙MP[2;⍙AC]]←⍙MP[2;⍙AC]↑⍕∊MAT[⍙AR;⍙AC]  ⍝ if bulk
⍝                                                               ⍝ else
Δ7:⎕WA←PlaceEntry ⍙AC                                           ⍝ adjust
Δ8:→0,RC←CFLAG←1
⍝
ΔE:→Δ1,0ρCs←11,1,0ρCs←2,5,0ρDs←'Invalid input'                  ⍝ error
```

```
RC←GetFieldType;TYP;DEC;I;T
⍝
⍝  Get field types of a new matrix
⍝  RC: 1 - ok,  0 - escape
⍝
 T←''
Δ1:T←T Input 'Enter field types ( C / D / Nx / Ex / A )'
 →(27=(I←1)↑T)ρRC←0
 →(2≤ρρT)ρΔE
 T←Parse T
 ⍙MP←4 0ρ⍳0
Δ2:→(ParseType I⊃T)↓ΔE
 ⍙MP←⍙MP,[2](3+(DEC×DEC≠99)+7×TYP=2),TYP,('CND')[TYP+1],DEC
 →((ρT)≥I←I+1)ρΔ2
 ⍙MA←(ρMAT←(¯2↑1,1,ρT)ρ((' '),MISSΔN,MISSΔN)[⍙MP[2;]+1])ρ0
 ⍙MP←(0,+\2+⍙MP[1;]),[1]⍙MP,3,0,0,0
 →0,ρ(⍙LC ⍙LR ⍙LS ⍙UR ⍙UC RC)←(,' ')((ρT)ρ' ')(' ')2 6 1
⍝
⍝  Error handling
ΔE:→Δ1,0ρCs←11,1,0ρCs←2,5,0ρDs←'Invalid input'
```

```
FILE←GetFileName FILE;EC
⍝
⍝  Get File name for write operations, check for existing file
⍝
Δ1:→Δshare↓ΔE
 →(27=1↑FILE←FILE Input 'Enter DOS file name')ρΔ3
 →(0≠EC←Δropen FILE,',D')ρΔ3
 Δclose
 →(GetYN 'File already exists, delete it')↓Δ1
 →Δshare↓ΔE
 →(0=EC←Δdelete FILE)ρΔ3
 →Δ2,0ρCs←11,1,0ρCs←2,5,0ρDs←'AP210 error: ',⍕ole[|EC;]
ΔE:Cs←11,1,0ρCs←2,5,0ρDs←'     AP210 not active'
Δ2:FILE←,27
Δ3:Δclose
```

```
NEW←GetInName
⍝
⍝  Get name of new imported matrix
⍝
Δ1:→(27=1↑NEW←GetName '')ρ0
 ⍎(2=⎕NC NEW)/'→(GetYN NEW,'' already exists; overwrite it'')↓Δ1'
 '→Δ1,ρCs←11,1,0ρCs←2,5,0ρDs←''Invalid name''' ⎕EA '→⍎NEW,''←0'''
```

```
NAME←GetName NAME
⍝
⍝  NAME=''  get and check name of new APL matrix
⍝  NAME≠''  check name
⍝  Returns the name if ok,  27 if Escape
⍝
 →(0<ρNAME)ρΔ2
 NAME←''
Δ1:→(27=1↑NAME←NAME Input 'Enter name of the new APL matrix')ρ0
Δ2:→(3≠⎕NC NAME)ρ0
⍝
⍝  Error handling
 →Δ1,ρCs←11,1,0ρCs←2,5,0ρDs←NAME,' is a function'
```

```
R←GetTitle C;S
⍝
⍝  Create title for submatrices
⍝
 →(' '≡⊃⍙LR[C])ρΔ1
 R←(8⌊ρR)↑R←⊃⍙LR[C]
 →((⎕AF R[1])∊S←(64+⍳26),(96+⍳26),182,247)↓Δ1     ⍝ check for a valid
 →(∧/∨/(⎕AF 1↓R)∘.∊S,95,253,47+⍳10)ρ0            ⍝ APL object name
⍝
Δ1:R←'C',⍕C                                   ⍝ use column number as name
```

```
RC←GetYN TXT;K
⍝
⍝  Ask for confirmation with prompt text TXT
⍝  RC: 1 - yes,   0 - no
⍝
 Cs←11,1,0ρCs←2,5,0ρDs←TXT,' (Y/N) ?'                   ⍝ display text
 →(∧/78 89 110 121≠K←1↑Ds,0ρCs←3,1)/⎕LC                 ⍝ get answer Y/N
 RC←(78 110 89 121=K)/0 0 1 1                           ⍝ set return code
```

```
RC←Help;ΔA;ΔB;ΔP
⍝
⍝  Popup help screen, adapted from EDIT2; RC: dummy
⍝
 →(RC←2=⎕NC '⍙help')/Δ1
 →0,0ρCs←11,1,0ρCfS←2,5,0ρDs←'Helpscreen not available'
⍝  Hide cursor
Δ1:Cs←12,0,0ρCs←12,0ρDs←6 1 1
⍝  Format help field
 Cs←1 7,0ρDs←3 7 19 43 2,ΔB←16×⌊⍙C[1;1]÷16
 Ds←(,(18 42ρ⍙C[1;20]),ΔB),(ΔB+⌊⍙C[1;20]÷16),42ρ16×⌊⍙C[1;20]÷16
 ΔP←0,0ρCs←7 7
Δ2:Ds←(42↑' UEdit        Help screen        Esc:Quit'),'▄',(42ρ'-'),'█'
 Cs←2 7,0ρDs←Ds,(,(16 42↑(16×ΔP)↓[1]⍙help),'█'),'▀',(41ρ'▄'),'█'
⍝  Esc, PgUp, PgDn are ok
 →(∧/27 1=ΔB←2↑Ds,0ρCs←3 1)/0
 →((0=∨/ΔA),ΔA←73 81=ΔB←2⊃ΔB)/Δ2,2ρΔ3
Δ3:→Δ2,ΔP←0⌈(¯1+⌈(↑ρ⍙help)÷16)⌊ΔP+((73=ΔB)/¯1),(81=ΔB)/1
```

```
RC←High F;K;R;C
⍝
⍝  (Un-)Highlight areas, (RC: 1-changed, 0-escape)
⍝  F: 1-field(short), 2-field, 3-block, 4-row, 5-column, 6-unhighlight
⍝
 →(∨/⍙AC ⍙AR>⍙MC ⍙MR)ρRC←0                           ⍝ cursor on valid field ?
 Cs←2,5,0ρDs←'     Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
 →F⊃Δ1,Δ2,Δ3,Δ4,Δ5,Δ7
⍝
Δ1:→Δ6,ρ(R C K)←(HIGHR HIGHC HIGHA)←⍙AR ⍙AC(10|¯1+1↓⊃⍙ktyp[PRESS])
Δ2:→(27=1↑K←GetColor)ρ0
 →Δ6,ρ(R C K)←(HIGHR HIGHC HIGHA)←⍙AR ⍙AC K
Δ3:→(HIGHR=0)ρ0
 →Δ6,ρ(R C K)←((HIGHR⌊⍙AR)+0,⍳|HIGHR-⍙AR)((HIGHC⌊⍙AC)+0,⍳|HIGHC-⍙AC)HIGHA
Δ4:→(27=1↑K←GetColor)ρ0
 →Δ6,ρ(R C)←⍙AR(⍳⍙MC)
Δ5:→(27=1↑K←GetColor)ρ0
 (R C)←(⍳⍙MR)⍙AC
Δ6:⍙MA[R;C]←K+8×⌊⍙MA[R;C]÷8
 ⍙DA[R;∊GC¨C]←(2+⍙MP[2;C])/((ρ,R),ρ,C)ρ⎕AF ⍙C[1;1+⍙MA[R;C]]
 ⍙UL[3]←(1+∨/0<8|,⍙MA)⊃' ' 'H'
 ClearUL
 →0,RC←CFLAG←1
⍝
Δ7:⍙MA←8×⌊⍙MA÷8                                         ⍝ Un-highlight
```

```
[ 25]  ⍙DA←(¯1↓2+⍙MP[2;])/⎕AF ⍙C[1;1+⍙MA]
[ 26]  ⍙UL[3]←' '
[ 27]  ClearUL
[ 28]  RC←CFLAG←1+HIGHR←HIGHC←0
```

```
[  0] R←HorShift
[  1] ⍝
[  2] ⍝  Adjust column labels, field types/widths, sets DIST, returns DISB
[  3] ⍝
[  4]  Cs←1,1,0⍴Ds←1,⍙UC,(⍙UR-1),⍙WC,2,⍙C[1;16]             ⍝ re-define field
[  5]  Cs←4,1,0⍴Ds←DIST←,((⍙UR-1),⍙WC)↑⍙MP[1;⍙OC←⍙CO]↓[2]⍙CL  ⍝ display labels
[  6]  Cs←4,3,0⍴Ds←R←⍙SC↑((⍙UC-1)⍴' '),⍙MP[1;⍙CO]↓⍙CT        ⍝ display formats
```

```
[  0] RC←InCSV;F;IN;R;L;NEW
[  1] ⍝
[  2] ⍝  Read comma delimited file (CSV format) into nested APL matrix
[  3] ⍝  Fields are expected to be enclosed in double quotes (⍝) if they
[  4] ⍝  contain commas.  RC: 1 - ok,  0 - escape
[  5] ⍝
[  6]  →Quit↓RC←0                    ⍝ Current matrix changed since last save ?
[  7]  F←''
[  8] ∆1:→(27=1↑F←F Input 'Enter CSV file name (with path if necessary)')⍴0
[  9]  Cs←2,5,0⍴Ds←'     Reading ',F,'...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 10]  →(0=⍴⍴IN←∆fv F)⍴0,⍴Cs←7,5,0⍴Ds←⍙C[1;18]
[ 11]  →(27=1↑NEW←GetInName)⍴0
[ 12]  Cs←(R←2),5,0⍴Ds←'     Parsing line 1 of ',⍕L←⍴IN      ⍝ Parse file
[ 13]  MAT←(1,⍴MAT)⍴MAT←Parse⊃IN[1]                         ⍝   into nested
[ 14] ∆3:Cs←2,5,0⍴Ds←'     Parsing line ',(⍕R),' of ',⍕L     ⍝   APL2 matrix
[ 15]  →(L≥R←R+1)⍴∆3,⍴MAT←MAT,[1]Parse⊃IN[R]
[ 16] ⍝
[ 17]  ⍎(ORIG←NEW),'←MAT'                                   ⍝ Save matrix in ws
[ 18]  RC←¯1+MakeParam                                      ⍝ Restart session
[ 19]  CFLAG←0
```

```
[  0] RC←InDOS;F;LEN;IN;IN0;R;C;NEW
[  1] ⍝
[  2] ⍝  Read DOS file into nested APL matrix
[  3] ⍝  RC: 1 - ok,  0 - escape
[  4] ⍝
[  5]  →Quit↓RC←0                     ⍝ Current matrix changed since last save ?
[  6]  LEN←F←''
[  7] ∆1:→(27=1↑F←F Input 'Enter DOS file name (with path if necessary)')⍴0
[  8]  Cs←2,5,0⍴Ds←'     Reading ',F,' ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[  9]  →(0=⍴⍴IN←⊃∆fv F)⍴0,⍴Cs←7,5,0⍴Ds←⍙C[1;18]
[ 10]  →(27=1↑NEW←GetInName)⍴0
[ 11] ∆3:→(27=1↑LEN←LEN Input 'Enter field lengths')⍴0
[ 12]  '→∆E2' ⎕EA '→((1↓⍴IN)<+/LEN←⍎LEN)⍴∆E3'
[ 13] ⍝
[ 14]  MAT←((R←(C←1)↑⍴IN0←IN),0)⍴⍳0
[ 15] ∆4:Cs←2,5,0⍴Ds←'     Parsing column ',(⍕C),' of ',⍕⍴LEN
[ 16]  MAT←MAT,Dtb¨⊂[2]LEN[C]↑[2]IN0
[ 17]  →((⍴LEN)≥C←C+1)⍴∆4,⍴IN0←LEN[C]↓[2]IN0
[ 18] ⍝
[ 19]  ⍎(ORIG←NEW),'←MAT'
[ 20]  RC←¯1+MakeParam                               ⍝ Create matrix parameters
[ 21]  →CFLAG←0                                      ⍝ Restart session
[ 22] ⍝
[ 23] ⍝  Error handling
[ 24] ∆E2:→∆3,⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Numeric data only, please'
[ 25] ∆E3:→∆3,⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Error in field lengths'
```

```
[  0] RC←InitPrt;X;A
[  1] ⍝
[  2] ⍝  INIT  --  Initialize HP LaserJet  (RC always 0)
[  3] ⍝  Reset HP LaserJet and load down APL2 font.
[  4] ⍝  The font will be permanent, so that it is not deleted from
[  5] ⍝  printer memory by a following reset.
[  6] ⍝
[  7]  ⎕WA←81 ⎕SVO 'X'
[  8]  →(2≠⎕SVO 'X')⍴∆Er
[  9] ⍝
[ 10] ⍝  Reset printer, download fonts, initialize
[ 11]  Cs←2,5,0⍴Ds←'     Downloading ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 12]  X←2,0⍴X←0
[ 13]  A←X
[ 14]  ⎕WA←⎕SVR 'X'
[ 15]  ⍎(2∣A)/'Cs←11,1,0⍴Cs←2,5,0⍴Ds←''Portrait font AP100RFN.SFP not found'''
[ 16]  ⍎(2≤A)/'Cs←11,1,0⍴Cs←2,5,0⍴Ds←''Landscape font AP100RFN.SFL not found'''
[ 17]  →RC←0
[ 18] ⍝
[ 19] ∆Er:⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'     Unable to share AP81'
```

```
[  0] R←CLR Input MSG;C
[  1] ⍝
[  2] ⍝  Prepare screen for user input; clear input line if CLR empty,
[  3] ⍝  otherwise fill input line with CLR and locate cursor behind it,
[  4] ⍝  display start message, get user input
[  5] ⍝
[  6]  Cs←7,5,0⍴Ds←⍙C[1;18]                      ⍝ Set message line to no-blink
[  7]  Cs←2,5,0⍴Ds←((⍙SC-17)↑MSG),'Hit Esc to cancel'     ⍝ Update status line,
[  8]  Cs←2,6,0⍴Ds←CLR                                    ⍝   open input line
[  9]  Cs←1,6,0⍴Ds←(⍙SR-2),1 1,⍙SC,0,⍙C[2;8]
[ 10] ⍝
[ 11] ∆1:Ds←6,1,↑1+(~(∊CLR)≡,' ')↓0,⍴,CLR                  ⍝ Get user input
[ 12] ∆2:Cs←3,0,0⍴Cs←12,0,0⍴Cs←12
[ 13]  ⍎(∧/4 1=C←2↑Ds)/'→∆3,R←27'                        ⍝ <Escape>
[ 14]  ⍎(∧/6 1=C)/'→∆2,Ds←6,1,1'                          ⍝ <Home>
[ 15]  ⍎(∧/6 2=C)/'→∆1,CLR←(Dtb,Ds),0⍴Cs←5,6'            ⍝ <End>
[ 16]  →(∨/0 0≠C)⍴∆2                                      ⍝ not <Return>
[ 17]  R←(Dtb,Ds),0⍴Cs←5,6
[ 18] ∆3:Cs←1,6,0⍴Ds←(⍙SR-2),1 1,⍙SC,2,⍙C[1;8]             ⍝ Close input line
[  0] RC←InsCol;TYP;DEC;C;T;MF;I
[  1] ⍝
[  2] ⍝  Insert blank columns before the current column
[  3] ⍝  RC: 1 - ok, 0 - escape
[  4] ⍝
[  5]  T←''
[  6] Δ1:T←T Input 'Enter field types ( C / D / Nx / Ex / A )'
[  7]  →(27=(I←1)↑T)ρRC←0
[  8]  →(2≤ρρT)ρΔE1
[  9]  Cs←2,5,0ρDs←'     Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[ 10]  T←Parse T
[ 11]  MF←5 0ρ⍳0
[ 12] Δ2:→(ParseType I⊃T)↓ΔE1
[ 13]  MF←MF,[2]0,(3+(DEC×DEC≠99)+7×TYP=2),TYP,('CND')[TYP+1],DEC
[ 14]  →((ρT)≥I←I+1)ρΔ2
[ 15]  T←↑ρT
[ 16]  ⍙LR←((⍙AC-1)↑⍙LR),(Tρ' '),(⍙AC-1)↓⍙LR
[ 17]  MAT←((⍙AC-1)↑[2]MAT),((⍙MR,T)ρ((' '),MISSΔN,MISSΔN)[MF[3;]+1]),(⍙AC-1)
         ↓[2]MAT
[ 18]  ⍙MA←((⍙AC-1)↑[2]⍙MA),((⍙MR,T)ρ0),(⍙AC-1)↓[2]⍙MA
[ 19]  ⍙DM←(C↑[2]⍙DM),((⍙MR,+/2+MF[2;])ρ' '),(C←⍙MP[1;⍙AC])↓[2]⍙DM
[ 20]  ⍙DA←(C↑[2]⍙DA),((⍙MR,+/2+MF[2;])ρ⎕AF ⍙C[1;1]),C↓[2]⍙DA
[ 21]  ⍙CL←(C↑[2]⍙CL),((2,+/2+MF[2;])ρ' '),C↓[2]⍙CL
[ 22]  ⍙CT←(C↑⍙CT),(∊(2+MF[2;])↑¨MF[4;],¨(⍕¨MF[2;]),¨(' ','.',¨⍕¨MF[5;])[1+
         (MF[5;]≠99)×⍳T]),C↓⍙CT
[ 23] ⍝
[ 24]  ⍙MP←((⍙AC-1)↑[2]⍙MP),MF,(⍙AC-1)↓[2]⍙MP
[ 25]  ⍙MP[1;]←0,¯1↓+\2+⍙MP[2;]
[ 26]  ⍙MC←⍙MC+T
[ 27]  ⍙CL[RC←CFLAG←1;]←∊(2+¯1↓⍙MP[2;])↑¨⍕¨⍳⍙MC
[ 28]  →0,DISB←HorShift
[ 29] ⍝
[ 30] ΔE1:→Δ1,0ρCs←11,1,0ρCs←2,5,0ρDs←'Invalid format'
```

```
[  0] RC←InsRow
[  1] ⍝
[  2] ⍝  Insert a blank row before the current row;  RC: always 1
[  3] ⍝
[  4]  Cs←2,5,0ρDs←'     Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[  5]  MAT←((⍙AR-1)↑[1]MAT),[1]((' '),MISSΔN,MISSΔN)[1+¯1↓⍙MP[3;]],[1](⍙AR-1)
         ↓[1]MAT
[  6]  ⍙DM←(((⍙AR-1)↑[1]⍙DM),[1]' '),[1](⍙AR-1)↓[1]⍙DM
[  7]  ⍙MA←((⍙AR-1)↑[1]⍙MA),[1]0,[1](⍙AR-1)↓[1]⍙MA
[  8]  ⍙DA←((⍙AR-1)↑[1]⍙DA),[1](⎕AF ⍙C[1;1]),[1](⍙AR-1)↓[1]⍙DA
[  9]  ⍙LC←((⍙AR-1)↑⍙LC),(' '),(⍙AR-1)↓⍙LC
[ 10]  DISL←VerShift
[ 11]  ⍙MR←⍙MR+CFLAG←RC←1
[  0] RC←LeMarg;L;V
[  1] ⍝
[  2] ⍝  Set left margin
[  3] ⍝
[  4]  L←⍕⍙pvar[4]
[  5] ∆1:→(27=1↑L←L Input 'Enter new left margin')⍴RC←0
[  6]  '→∆Er' ⎕EA '⍙pvar[4]←V←999⌊⌊|⍎L'
[  7]  →0,⍴⍙menu3[6;27+⍳3]←'550'⍕V
[  8] ⍝
[  9] ∆Er:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Input must be numeric'
```

```
[  0] RC←LinInch;L;V
[  1] ⍝
[  2] ⍝  Set lines per inch
[  3] ⍝
[  4]  L←⍕⍙pvar[7]
[  5] ∆1:→(27=1↑L←L Input 'Enter lines per inch')⍴RC←0
[  6]  '→∆Er' ⎕EA '⍙pvar[7]←V←⌊48÷1⌈⌈48÷|⍎L'
[  7]  ⍙menu3[9;27+⍳3]←'550'⍕V
[  8]  →0,⍴⍙menu3[8;27+⍳3]←'550'⍕⍙pvar[6]←⌈(0.2×V)⌈⍙pvar[6]
[  9] ⍝
[ 10] ∆Er:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Input must be numeric'
```

```
[  0] RC←Locate;R
[  1] ⍝
[  2] ⍝  Position cursor at specific row and column
[  3] ⍝
[  4]  R←''
[  5] ∆1:→(27=1↑R←R Input 'Enter row,column')⍴RC←0
[  6]  '→∆E1' ⎕EA '→(2≠⍴R←⍎R)⍴∆E2'                ⍝ must be numeric, 2 elements
[  7]  ⍙CO←((1+⍙SC-⍙UC)≥20+(+/⍙MP[1 2;⍙AC←1⌈⍙MC⌊R[2]])-⍙MP[1;])⍳1
[  8]  ⍙RO←1⌈¯5+⍙AR←1⌈⍙MR⌊R[1]
[  9]  ⍎(⍙CO≠⍙OC)/'DISB←HorShift'                  ⍝ scroll window horizontally
[ 10]  ⍎(⍙RO≠⍙OR)/'DISL←VerShift'                  ⍝ and vertically if necessary
[ 11]  →0,RC←1
[ 12] ⍝
[ 13] ⍝  Error handling
[ 14] ∆E1:→∆E0,MSG←'Input must be a numeric vector'
[ 15] ∆E2:MSG←'Specify 2 columns, please'
[ 16] ∆E0:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG
```

```
[  0] R←MakeDisA;C
[  1] ⍝
[  2] ⍝  Build attribute matrix for display window
[  3] ⍝
[  4]  C←⎕AF((⍙WR⌊⍙MR-R),⍙WC⌊⍙MP[1;⍙MC+1]-C)↑((R←⍙RO-1),C←⍙MP[1;⍙CO])↓⍙DA
[  5]  R←(⍙WR,⍙WC)⍴⍙C[1;1]
[  6]  R[⍳1↑⍴C;⍳1↓⍴C]←C
[  7]  R←,R
[  0] R←MakeDisP
[  1] ⍝
[  2] ⍝  Cut window portion from display matrix
[  3] ⍝
[  4]  R←,(⍙WR,⍙WC)↑((⍙RO-1),⍙MP[1;⍙CO])↓⍙DM
```

```
[  0] R←P MakeIndex A;W;C;D;I;U
[  1] ⍝
[  2] ⍝  Extract unique elements for FreqTab and CrossTab
[  3] ⍝  I      input vector A converted to indices
[  4] ⍝  U      unique elements, used as col/row labels
[  5] ⍝  P[1]   column number
[  6] ⍝  P[2]   0 - A is a vector of character elements
[  7] ⍝         else: classification data (L,U,×classes) for numeric vector A
[  8] ⍝
[  9]  (C D)←P
[ 10] ⍝  Character vector
[ 11]  ⍎(0=⍴⍴D)/'→∆2,I←(U←U[(((⍴U),1)⍴U←((A⍳A)=⍳⍴A)/A)sort∆sub 1])⍳A'
[ 12] ⍝
[ 13] ⍝  Else
[ 14]  I←1⌈(2+⌊(A-D[1])÷W←(D[2]-D[1])÷D[3])⌊D[3]+2
[ 15]  →(2=⍙MP[3;C])⍴∆1
[ 16] ⍝
[ 17] ⍝  Numeric vector
[ 18]  →∆2,⍴U←⊂[2]('<',(D[3]+1)⍴'≥'),' ',⍕((D[3]+2),1)⍴D[1]+W×0,0,⍳D[3]
[ 19] ⍝
[ 20] ⍝  Date vector
[ 21] ∆1:U←⊂[2]('<',(D[3]+1)⍴'≥'),' ',FromDays D[1]+W×0,0,⍳D[3]
[ 22] ⍝
[ 23] ∆2:R←(I U)
```

```
[  0] RC←MakeLabCol;C
[  1] ⍝
[  2] ⍝  Toggle 1st matrix column: column ←→ label;  RC dummy
[  3] ⍝
[  4]  Cs←2,5,0⍴Ds←'      Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[  5]  →(⍙UC=6)⍴∆1,CFLAG←RC←1
[  6] ⍝
[  7]  (MAT ⍙MA)←(⍙LC,MAT)(0,⍙MA)                              ⍝ 1st column ← label
[  8]  ⍙DM←((C←2+1↓⍴⊃⍙LC)↑[2]⊃⍙LC),⍙DM
[  9]  ⍙DA←(((↑⍴⍙DA),C)⍴⎕AF ⍙C[1;1]),⍙DA
[ 10]  (⍙CL ⍙CT)←((C↑[2]' ',[0.5],⊃⍙LS),⍙CL)((C↑'C',⍕C-2),⍙CT)
[ 11]  (⍙LC ⍙LS ⍙LR ⍙UC ⍙MC)←(⍙MR⍴' ')(' ')(⍙LS,⍙LR)6(⍙MC+1)
[ 12]  →∆2,⍴⍙MP←(0,(C-2),0,'C',99),[2]⍙MP
[ 13] ⍝
[ 14] ∆1:⍎(0≠⍙MP[3;1])/'⎕WA←''C'' ToggType 1'                  ⍝ label ← 1st column
[ 15]  (⍙LC ⍙LR ⍙LS)←(MAT[;1])(1↓⍙LR)(1↑⍙LR)
[ 16]  ⍎(1=≡⍙LC)/'⍙LC←⊂[2](((⍴⍙LC),1)⍴⍙LC),'' '''
[ 17]  (MAT ⍙MA)←1↓[2]¨MAT ⍙MA
[ 18]  (⍙DM ⍙DA ⍙CL)←⍙MP[1;2]↓[2]¨⍙DM ⍙DA ⍙CL
[ 19]  (⍙UC ⍙CT ⍙MC)←(7+↑1↓⍴⊃⍙LC)(⍙MP[1;2]↓⍙CT)(⍙MC-1)
[ 20]  ⍙MP←1↓[2]⍙MP
[ 21] ⍝
```

```
Δ2:⍙MP[1;]←0,¯1↓+\2+⍙MP[2;]                      ⍝ adjust column positions
 ⍙CL[1;]←∊(2+¯1↓⍙MP[2;])↑¨(⍕¨⍳⍙MC)
 ⍙WC←1+⍙SC-⍙UC
```

```
RC←MakeLabRow;NoNum
⍝
⍝  1st matrix row → column labels if no labels exist
⍝  column labels → 1st matrix row otherwise;  RC is dummy
⍝
 Cs←2,5,0ρDs←'     Working ...',0ρCs←7,5,0ρDs←⍙C[2;18]
 →(RC←⍙UR=2)ρΔ1
⍝
⍝  1st row ← labels
 NoNum←⍳0
 ChgType¨(0≠¯1↓⍙MP[3;])/⍳⍙MC
 (MAT ⍙MA ⍙DM)←(⍙LR,[1]MAT)(0,[1]⍙MA)(⍙CL[2;],[1]⍙DM)
 ⍙DA←(⎕AF ⍙C[1;1]),[1]⍙DA
 ⍙LR←⍙MCρ' '
 ⍙CL[2;]←' '
 ⍎(0<ρNoNum)/'Cs←11,1,0ρCs←2,5,0ρDs←''Columns'',(⍕NoNum),'' set to
    character type'''
 (⍙MR ⍙LS ⍙LC)←(⍙MR+1)(' ')(⍙LS,⍙LC)
 →0,⍙WR←⍙SR-3+⍙UR←2
⍝
⍝  label ← 1st row
Δ1:(⍙LR ⍙MR)←(⍕¨MAT[1;])(⍙MR-1)
 ⍙CL[2;]←⍙DM[1;]
 (MAT ⍙MA ⍙DM ⍙DA)←1↓[1]¨MAT ⍙MA ⍙DM ⍙DA
 (⍙LS ⍙LC)←(1↑⍙LC)(1↓⍙LC)
 ⍙WR←⍙SR-3+⍙UR←3
```

```
RC←MakeParam;I;COL;W;N;D;C
⍝
⍝  Generate matrix parameters for a new matrix
⍝     RC       always 2
⍝     ⍙MP[1;]  screen column offset
⍝     ⍙MP[2;]  field (=column) lengths
⍝     ⍙MP[3;]  field types (0-character, 1-numeric, 2-date)
⍝     ⍙MP[4;]  field types (C-character, N-numeric, D-date)
⍝     ⍙MP[5;]  decimals for fixed-point numeric data, 99 otherwise
⍝
 Cs←2,5,0ρDs←'     Creating parameters ...',0ρCs←7,5,0ρDs←⍙C[2;18]
 ⍙MA←⍎⊃('(ρMAT)ρ0' 'RATT')[1+2=⎕NC 'RATT']
 (⍙LC ⍙LR ⍙LS ⍙UR ⍙UC RC)←((↑ρMAT)ρ' ')((1↓ρMAT)ρ' ')(' ')2 6 2
⍝
⍝  Check for column labels in 1st row
 →(∨/(1=↑ρMAT),∧/¨MAT[1;]∊¨N←⊂'¯.0123456789E')ρΔ1        ⍝ No column labels
 (⍙LC ⍙LR ⍙UR)←(1↓⍙LC)(MAT[1;])3                      ⍝ Col labels exist
 (MAT ⍙MA)←1↓[1]¨MAT ⍙MA
⍝
⍝  Check for row labels in 1st column
Δ1:→(∨/(1=1↓ρMAT),∧/¨MAT[;1]∊¨N)ρΔ2                      ⍝ No row labels
 ⍙LC←MAT[;1]                                          ⍝ Row labels exist
 ⍎(1=≡⍙LC)/'⍙LC←⊂[2](((⍴⍙LC),1)⍴⍙LC),'' '''
 ⍙UC←7+↑1↓⍴⊃⍙LC
 (⍙LR ⍙LS)←(1↓⍙LR)(1↑⍙LR)
 (MAT ⍙MA)←1↓[2]¨MAT ⍙MA
⍝
⍝  Create parameters
Δ2:(⍙MR ⍙MC)←⍴MAT
 I←↑1,⍴⍙MP←4 0⍴⍙CL←2 0⍴⍙DM←(⍙MR,0)⍴⍙CT←⍳0
Δ3:→(∧/0<N←∊ToDays¨COL←MAT[;I])⍴Δ5                    ⍝ Is it a date column ?
 '→Δ4' ⎕EA 'COL←MAT[;I]←⍎¨MAT[;I]'                     ⍝   or numeric ?
⍝
 COL[N←(MISSΔN=COL)/⍳⍴COL]←0                            ⍝ numeric column
 COL←⍕(⍙MR,1)⍴COL
 COL[N;]←' '
 W←2+3⌈(↑⍴⊃⍙LR[I])⌈¯1↑⍴COL
 ⍙MP←⍙MP,[2](W-2),1,'N',99
 →Δ7,⍴⍙DM←⍙DM,[2]W↑[2]COL
⍝
Δ4:W←2+3⌈(↑⍴⊃⍙LR[I])⌈⌈/∊⍴¨COL                          ⍝ Character column
 →Δ6,⍴⍙MP←⍙MP,[2](W-2),0,'C',99
⍝
Δ5:MAT[;I]←N                                           ⍝ date column
 ⍙MP←⍙MP,[2](W←10),2,'D',99
⍝
Δ6:⍙DM←⍙DM,[2]W↑[2](2↑(⍴COL),1)⍴COL←⊃COL
Δ7:⍙CL←⍙CL,(2,W)⍴(W↑⍕I),W↑I⊃⍙LR
 ⍙CT←⍙CT,W↑(,⍙MP[3;I]),(⍕W-2),,⊃(⍙MP[4;I]≠99)/⊂'.',⍕⍙MP[4;I]
 →(⍙MC≥I←I+1)⍴Δ3
⍝
 ⍙MP←(0,+\2+⍙MP[1;]),[1]⍙MP,3,0,0,0                   ⍝ Position of matrix end
 SetPtr


RC←Mark F;R;C
⍝
⍝  (Un-)Mark areas, (RC: 1-dummy)
⍝  F: 1-field, 2-block, 3-row, 4-column, 5-unhighlight
⍝  'mark field' toggles, block/row/column set marks
⍝
 →(∨/⍙AC ⍙AR>⍙MC ⍙MR)⍴RC←0                           ⍝ cursor on valid field ?
 Cs←2,5,0⍴Ds←'      Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 →F⊃Δ1,Δ2,Δ3,Δ4,Δ6
⍝
Δ1:⍙MA[⍙AR;⍙AC]←(8×⍙MA[⍙AR;⍙AC]<8)+8|⍙MA[⍙AR;⍙AC]          ⍝ Mark field
 ⍙DA[⍙AR;GC ⍙AC]←(2+⍙MP[2;⍙AC])/⎕AF ⍙C[1;1+⍙MA[⍙AR;⍙AC]]
 ⍙UL[1]←'M'
 ClearUL
 →0,(CFLAG MARKR MARKC)←1 ⍙AR ⍙AC
⍝
Δ2:→(MARKR=0)⍴0                                            ⍝ Mark block
 →Δ5,⍴(R C)←((MARKR⌊⍙AR)+0,⍳|MARKR-⍙AR)((MARKC⌊⍙AC)+0,⍳|MARKC-⍙AC)
Δ3:→Δ5,⍴(R C)←⍙AR(⍳⍙MC)                                    ⍝ Mark row
Δ4:(R C)←(⍳⍙MR)⍙AC                                         ⍝ Mark column
Δ5:⍙MA[R;C]←8+8|⍙MA[R;C]
 ⍙DA[R;∊GC¨C]←(2+⍙MP[2;C])/((⍴,R),⍴,C)⍴⎕AF ⍙C[1;1+⍙MA[R;C]]
```

```
[ 22]  ∆UL[1]←'M'
[ 23]  ClearUL
[ 24]  →0,CFLAG←1
[ 25] ⍝
[ 26] ∆6:∆MA←8|∆MA                                                  ⍝ Unmark
[ 27]  ∆DA←(¯1↓2+∆MP[2;])/⎕AF ∆C[1;1+∆MA]
[ 28]  ∆UL[1]←' '
[ 29]  ClearUL
[ 30]  CFLAG←1+MARKR←MARKC←0


[  0] RC←Menu NUM;∆B;MSC
[  1] ⍝
[  2] ⍝  Popup menu for additional functions
[  3] ⍝  NUM:  1  Statistics              RC:  1 screen update necessary
[  4] ⍝        2  File Operations              0 escape
[  5] ⍝        3  Printer
[  6] ⍝
[  7]  RC←0
[  8]  →(2=⎕NC MSC←'∆menu',⍕NUM)/∆1
[  9]  →0,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Menu not available'
[ 10] ⍝  Format menu field
[ 11] ∆1:Cs←1 12,0⍴Ds←6 40 13 32 2,∆B←16×⌊∆C[1;1]÷16
[ 12]  Cs←7 12,0⍴Ds←(,(12 31⍴∆C[1;20]),∆B),(∆B+⌊∆C[1;20]÷16),31⍴16×⌊∆C[1;20]÷16
[ 13]  Ds←(31↑(NUM⊃∆menu0),'      Esc:Quit'),'▄',(31⍴'-'),'█'
[ 14]  Cs←2 12,0⍴Ds←Ds,(,(10 31↑⍎MSC),'█'),'▀',(30⍴'▄'),'█'
[ 15] ⍝
[ 16] ∆2:→(∧/27 1=∆B←2↑Ds,0⍴Cs←3 1)⍴0                        ⍝ Escape pressed ?
[ 17]  →((∧/0 59≤∆B)∧(∧/0 68≥∆B))↓∆2                        ⍝ Function key ?
[ 18]  →(0=⎕NC 'DISA')⍴∆3
[ 19]  Cs←7,8,0⍴Ds←DISA                                      ⍝ edit window attributes
[ 20]  Cs←2,8,0⍴Ds←DISP                                      ⍝ edit window
[ 21] ⍝
[ 22] ∆3:⍎'RC←',⊃∆func[NUM;∆B[2]-58]                         ⍝ Execute function


[  0] RC←NewMat NAME
[  1] ⍝
[  2] ⍝  (Create and) Initialize a new matrix
[  3] ⍝  RC: 0 - escape (= keep old matrix)
[  4] ⍝      1 - display not updated
[  5] ⍝      2 - display matrix is correct
[  6] ⍝
[  7]  →Quit↓RC←0                                    ⍝ QQUIT old matrix ?
[  8]  CFLAG←0
[  9]  Cs←2,6,0⍴Ds←''                                ⍝ Get name of new matrix
[ 10]  →(27=1↑ORIG←GetName NAME)⍴0                   ⍝ Exit if escape
[ 11]  →(2=⎕NC ORIG)⍴∆1                              ⍝ Jump if matrix exists
[ 12]  ⍎(ReadMat ORIG)/'→0,RC←1'                     ⍝ Try to find file
[ 13] ⍝
[ 14]  →(GetFieldType)↓0                             ⍝ Get Field types of a new matrix
[ 15]  SetPtr
[ 16]  →0,RC←1
[ 17] ⍝
[ 18] ∆1:→(2=⎕NC '∆SP')⍴∆2                            ⍝ Statistics recursion ?
```

```
 ⍎(2≠⎕NC 'RATT')/'→(ReadMat ORIG)⍴0,RC←1' ⍝ Try to find file
 MAT←(¯2↑1,1,⍴⍎ORIG)⍴⍕¨⍎ORIG
 →0,RC←MakeParam                          ⍝ Create parameters for a matrix
∆2:→0,RC←StatParam                        ⍝ Recursion, matrix is ready
```

```
Z←Num X
⍝
⍝  Test if array X is numeric;  Z: 0 - no, 1 - yes
⍝
 Z←0∊0\0⍴X
```

```
RC←NumCol;L
⍝
⍝  Set number of columns
⍝
 L←⍕⍙pvar[5]
∆1:→(27=1↑L←L Input 'Enter number of columns')⍴RC←0
 '→Err' ⎕EA '⍙pvar[5]←999⌊1⌈⌊|⍎L'
 →0,⍴⍙menu3[7;27+⍳3]←'550'⍕⍙pvar[5]
⍝
∆Err:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Input must be numeric'
```

```
COL←NumToChar COL;N
⍝
⍝  Convert numeric column to character; C global column number
⍝
 COL[N←(MISS∆N=COL)/⍳⍴COL]←0
 ⍎'COL←',(,⊃(⍙MP[5;C]≠99)/⊂'1↓[2]⍙MP[5;C]'),'⍕(⍙MR,1)⍴COL'
 COL[N;]←' '
```

```
RC←OutCSV;FILE;N;OUTMAT;HMAT
⍝
⍝  Write current matrix to a CSV file;  RC: always 0
⍝
 FILE←''
 →(27=1↑FILE←GetFileName FILE)⍴RC←0
 HMAT←MAT
 N←(2=¯1↓⍙MP[3;])/⍳⍙MC
 ⍎(0≠⍴N)/'HMAT[;N]←(⍙MR,⍴N)⍴⊂[2]FromDays HMAT[;N]'
 HMAT←CompMat HMAT                                ⍝ Append labels
 OUTMAT←⍴0                                        ⍝ Prepare matrix as
 CSVprep¨⍳↑⍴HMAT                                  ⍝   vector of vectors
 Cs←2,5,0⍴Ds←'     Writing ',FILE,' ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 ⎕WA←OUTMAT ∆fv FILE
 Cs←7,5,0⍴Ds←⍙C[1;18]
```

```
RC←OutDOS;FILE;N;C1;D1;OUTMAT
⍝
⍝  Write current matrix to DOS file;  RC: always 0
⍝
 FILE←''
 →(27=1↑FILE←GetFileName FILE)⍴RC←0
 Cs←2,5,0⍴Ds←'     Preparing ',FILE,' ...'
 OUTMAT←⍙DM
 ⍎(⍙UR=3)/'OUTMAT←⍙CL[2;],[1]OUTMAT'
 ⍎(⍙UC>6)/'OUTMAT←(⊃((⍙UR=3)/⍙LS),⍙LC),'' '','' '',OUTMAT'
 N[('¯'=N)/⍳⍴N←,OUTMAT]←'-'
 Cs←2,5,0⍴Ds←'     Writing ',FILE,' ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 ⎕WA←(⊂[2]¯1↓[2](⍴OUTMAT)⍴N)Δfv FILE
 Cs←7,5,0⍴Ds←⍙C[1;18]
```

```
R←Parse CSV;Q;DEL;N
⍝
⍝  Convert a character vector into a nested vector
⍝  Commas are delimiters if not between double quotes
⍝
 DEL[N]←N←(DEL=1)/⍳⍴DEL←(~Q)/(CSV=',')∧~2|+\Q←'⍝'=CSV←',',CSV
 R←Dtb¨1↓¨(DEL+1)⊂(~Q)/CSV
```

```
RC←ParseType V
⍝
⍝  Create type parameters from user input
⍝  TYP  (global) type code
⍝  DEC  (global) decimal digits, 99 if character or date
⍝
 →('CcDdAaNnEe'⍳1↑V)⊃Δ1,Δ1,Δ2,Δ2,Δ3,Δ3,Δ4,Δ4,Δ5,Δ5,RC←0
Δ1:→0,⍴(RC TYP DEC)←1 0 99                         ⍝ character type
Δ2:→0,⍴(RC TYP DEC)←1 2 99                         ⍝ date
Δ3:→0,⍴(RC TYP DEC)←1 1 99                         ⍝ numeric (std.APL)
Δ4:'→0' ⎕EA '→0,⍴(RC TYP DEC)←1 1(⍎1↓V)'           ⍝ fixed decimals
Δ5:'→0' ⎕EA '→0,⍴(RC TYP DEC)←1 1(¯1-⍎1↓V)'        ⍝ scientific format
```

```
W←PlaceEntry C;COL;DIF;KEEPC
⍝
⍝  Rewrite matrix column C into ⍙DM, expand/reduce column width
⍝  Return:  W new column width
⍝
 Cs←2,5,0⍴Ds←'     Updating display ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 →(0=DIF←(W←DisCol)-2+⍙MP[2;C])⍴Δ2
 KEEPC←(1↓⍴⍙DM)⍴1
 ⍎(DIF<0)/'→Δ1,KEEPC[⍙MP[1;C]+W+⍳|DIF]←0'            ⍝ reduce width
 KEEPC[⍙MP[1;C+1]]←DIF+1                             ⍝ expand column
⍝
Δ1:(⍙DM ⍙DA ⍙CL ⍙CT)←KEEPC/¨⍙DM ⍙DA ⍙CL ⍙CT
 ⍙MP[1;]←0,¯1↓+\2+⍙MP[2;]
Δ2:⍙DM[;⍙MP[1;C]+⍳W]←W↑[2]COL
 ⍙DA[;⍙MP[1;C]+⍳W]←⍉(W,↑⍴⍙DA)⍴⎕AF ⍙C[1;1+⍙MA[;C]]
 ⍙CT[⍙MP[1;C]+⍳W]←W↑(,⍙MP[4;C]),(⍕W-2),,⊃(⍙MP[5;C]≠99)/⊂'.',⍕⍙MP[5;C]
```

```
Play;A;SH
⍝
⍝  Sound if matrix is ready to edit; length depends on matrix size
⍝
 A←440 ⎕SVO 'SH'
 ⍎(2≠⎕SVO 'SH')/'→0,⍴⎕←''NOT SHARED'''
 SH←'04',(2×⌈0.003×⍙MC×⍙MR)⍴'C6D6'
```

```
RC←Pool;D;COL;Cc;A;B;C2
⍝
⍝  Collapse specified rows/columns of a cross matrix
⍝
 →(0=⎕NC 'RCM')⍴RC←0                ⍝ return if no cross matrix exists
⍝
 Cs←11,1,0⍴Cs←2,5,0⍴Ds←'Collapse Rows or Columns; Original (R/C/O) ?'
 →(∧/67 79 82 99 111 114≠K←1↑Ds,0⍴Cs←3,1)/⎕LC     ⍝ get answer
 D←(67 99 82 114 79 111=K)/0 0 1 1 2 2           ⍝ cols=0, rows=1, orig=2
⍝
⍝  Back to the original table
 →(D≠2)⍴Δ0
 Cs←2,5,0⍴Ds←'     Restoring ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 →Δ5,⍴(RCM Uni1 Uni2)←ORCM OUni1 OUni2
⍝
⍝  Collapse rows/columns
Δ0:⍎(~D)/'RCM←⍉RCM'
 COL←''
⍝   Δ1:→(27=1↑COL←COL Input 'Enter row/column numbers')⍴0
Δ1:→(27=1↑COL←COL Input 'Enter ',(∊('column' 'row')[D+1]),' numbers')⍴0
 →(COL≡,' ')⍴0
 '→ΔE1' ⎕EA 'COL←⍎COL'                                ⍝ must be numeric
 ⍎D/'COL←1+⌊COL÷3'
 →(∧/COL∊⍳↑⍴RCM)↓ΔE2                                  ⍝ must exist
⍝
 Cs←2,5,0⍴Ds←'     Aggregating ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 RCM[↑COL;]←+⌿RCM[COL;]
 RCM←RCM[(⍳↑⍴RCM)~1↓COL;]
 →(~D)⍴Δ3
⍝
 →('<'≠↑,⊃Uni1[↑Cc←COL[⍋COL]])⍴Δ2                     ⍝ Rows
 '→ΔE3' ⎕EA 'Uni1[↑COL]←⊂''<'',1↓,⊃Uni1[1+¯1↑Cc]'
Δ2:→Δ5,⍴Uni1←Uni1[(⍳⍴Uni1)~1↓COL]
⍝
Δ3:→('<'≠↑,⊃Uni2[↑Cc←COL[⍋COL]])⍴Δ4                   ⍝ Columns
 '→ΔE3' ⎕EA 'Uni2[↑COL]←⊂''<'',1↓,⊃Uni2[1+¯1↑Cc]'
Δ4:Uni2←Uni2[(⍳⍴Uni2)~1↓COL]
 RCM←⍉RCM
⍝
Δ5:
 ⍎MM,'←CtPrep RCM'
 C2←CFLAG                          ⍝ temporarily disable the Change flag
 CFLAG←0
 D←NewMat MM
 Refresh
```

```
[ 45]  CFLAG←C2
[ 46]  →0,RC←1
[ 47] ⍝
[ 48] ⍝  Error handling
[ 49] ΔE1:→ΔE0,MSG←'Input must be a numeric vector'
[ 50] ΔE2:→ΔE0,MSG←'Non existing column/row numbers specified'
[ 51] ΔE3:MSG←'  ? ? ? ?'
[ 52] ΔE0:→Δ1,0ρCs←11,1,0ρCs←2,5,0ρDs←MSG
```

```
[  0] RC←PortLand;P
[  1] ⍝
[  2] ⍝  Toggle printer portrait/landscape mode
[  3] ⍝
[  4]  P←⍙pvar[3]←~⍙pvar[3]
[  5]  ⍙menu3[5;]←30↑' F5   Start ',((P+1)⊃'Landscape' 'Portrait'),' mode'
[  6]  ⍙pvar[1 2]←⍙pvar[2 1]
[  7]  RC←0
```

```
[  0] RC←PrtMark;R;C;MP;OMAT;N;LC
[  1] ⍝
[  2] ⍝  Print marked area of a worksheet
[  3] ⍝
[  4]  RC←0
[  5]  ⍎((⍳0)≡R←(∨/⍙MA≥8)/⍳⍙MR)/'→0,OMAT←'''''                 ⍝ marked rows
[  6]  C←(∨⌿⍙MA≥8)/⍳⍙MC                                        ⍝ marked columns
[  7] ⍝
[  8]  MP←⍙MP[2;C]
[  9]  OMAT←⍙DM[R;∊GC¨C]
[ 10]  ⍎(⍙UR=3)/'OMAT←(∊(2+⍙MP[2;C])↑¨⍙LR[C]),[1]OMAT'
[ 11]  ⍎(⍙UC>6)/'OMAT←(LC←⊃((⍙UR=3)/⍙LS),⍙LC[R]),'' '','' '',OMAT'
[ 12]  N[('¯'=N)/⍳ρN←,OMAT]←'-'
[ 13]  OMAT←(ρOMAT)ρN
[ 14]  ⍎(⍙UC>6)/'MP←(¯1↑ρLC),MP'
[ 15] ⍝
[ 16]  RC←PrtMat OMAT
```

```
[  0] RC←PrtMat OMAT;HD;R;C;RL;CL;RC;LPP;CPP;P;PP;PA;PB;PAG;MMP;L
[  1] ⍝
[  2] ⍝  Print OMAT which is the complete worksheet or a marked block
[  3] ⍝
[  4]  →(27=1↑HD←'' Input 'Enter heading for each page')ρRC←0
[  5]  →(27=1↑R←GetYN 'Row labels on each page')ρ0
[  6]  →(27=1↑C←GetYN 'Column labels on each page')ρ0
[  7]  Cs←2,5,0ρDs←'     Printing ...',0ρCs←7,5,0ρDs←⍙C[2;18]
[  8] ⍝
[  9]  LPP←⌊⍙pvar[7]×⍙pvar[1]-0.8+⍙pvar[6]÷⍙pvar[7]  ⍝ × matrix line per page
[ 10]  LPP←LPP-3+C×2                                 ⍝ Adjust for heading, labels
[ 11]  CPP←⍙pvar[5]-2+R×MP[1]                        ⍝ × matrix columns per page
[ 12] ⍝
[ 13]  MP←MP+((¯1+ρMP)ρ2),0                       ⍝ matrix field lengths
[ 14]  OMAT←¯2↓[2]OMAT                            ⍝ last 2 columns are blank
[ 15]  RL←((1↑ρOMAT),0)ρ''                        ⍝ init row labels
[ 16]  CL←(1↓ρOMAT)ρ' '                           ⍝ init column labels
[ 17]  RC←''                                      ⍝ init upper left label corner
[ 18]  →R↓Δ1
[ 19]  RL←MP[1]↑[2]OMAT                           ⍝ row labels
[ 20]  OMAT←MP[1]↓[2]OMAT
[ 21]  MP←1↓MP
[ 22] Δ1:→C↓Δ2
[ 23]  CL←OMAT[1;]                                ⍝ column labels
[ 24]  OMAT←1↓[1]OMAT
[ 25]  →R↓Δ2
[ 26]  RC←RL[1;]
[ 27]  RL←1↓[1]RL
[ 28] ⍝
[ 29] ⍝  Send printer setup codes
[ 30] Δ2:⎕WA←Δprint(⎕AF 27 38 108),(⍕⍙pvar[3]),'O'       ⍝ portrait/landscape
[ 31]  ⎕WA←Δprint(⎕AF 27 40),(⍕105+⍙pvar[3]),'X'        ⍝ set font id
[ 32]  ⎕WA←Δprint(⎕AF 27 38 97),(⍕⍙pvar[4]),'L'         ⍝ left margin
[ 33]  ⎕WA←Δprint(⎕AF 27 38 108),(⍕⍙pvar[6]),'E'        ⍝ top margin
[ 34]  ⎕WA←Δprint(⎕AF 27 38 108),(⍕⍙pvar[7]),'D'        ⍝ lines per inch
[ 35]  ⎕WA←Δprint(⎕AF 27 38 107 48 71)                  ⍝ line termination code
[ 36] ⍝
[ 37]  PB←0                                   ⍝ 2nd part of page number
[ 38] Δ3:P←MMP[PP←+/CPP≥MMP←+\MP]             ⍝ last column to print in this pass
[ 39]  PA←0                                   ⍝ 1st part of page number
[ 40]  PB←PB+L←1                              ⍝ L: matrix line to print
[ 41] ⍝
[ 42] Δ4:PA←PA+1
[ 43]  ⎕WA←Δprint((⍙pvar[5]-ρPAG)↑HD),(PAG←'page ',(⍕PA),'.',⍕PB),⎕AF 13 10 13 10
[ 44]  ⍎C/'⎕WA←Δprint (⎕AF 13 10),RC,CL[⍳P],⎕AF 13 10'
[ 45] ⍝
[ 46] Δ5:→((1↑ρOMAT)<L←L+1)ρΔ6,Δprint(⎕AF 13 10),RL[L;],OMAT[L;⍳P]
[ 47]  →(0≠LPP|L)ρΔ5
[ 48]  →Δ4,ff
[ 49] ⍝
[ 50] Δ6:OMAT←P↓[2]OMAT
[ 51]  CL←P↓CL
[ 52]  →(1≤ρMP←PP↓MP)ρΔ3,ff


[  0] RC←PrtWKS;OMAT;MP;N;LC
[  1] ⍝
[  2] ⍝  Print complete worksheet
[  3] ⍝
[  4]  OMAT←⍙DM
[  5]  ⍎(⍙UR=3)/'OMAT←⍙CL[2;],[1]OMAT'
[  6]  ⍎(⍙UC>6)/'OMAT←(LC←⊃((⍙UR=3)/⍙LS),⍙LC),'' '','' '',OMAT'
[  7]  N[('¯'=N)/⍳ρN←,OMAT]←'-'
[  8]  OMAT←(ρOMAT)ρN
[  9]  MP←¯1↓⍙MP[2;]
[ 10]  ⍎(⍙UC>6)/'MP←(¯1↑ρLC),MP'
[ 11] ⍝
[ 12]  RC←PrtMat OMAT
```

```
RC←Quit
⍝
⍝  Quit without saving changes ?
⍝  RC:  1 - ok to quit (do not save) or no changes applied
⍝       0 - do not quit
⍝
 RC←⍎(CFLAG+1)⊃'1' 'GetYN ''Lose all changes'''
```

```
RC←ReadMat F;FF;Cz;Dz;EC;MM
⍝
⍝  Read matrix and its parameters,  RC: 1 - ok, 0 - escape/error
⍝
 →Quit↓RC←0                          ⍝ Current matrix changed since last save ?
⍝
 ⍎(~F≡'')/'→Δ2,ρFF←PATH,F,''.UED,A'''
Δ1:→(27=1↑F←F Input 'Enter matrix name')ρRC←0
 FF←PATH,((∧\F≠'.')/F),'.UED,A'
Δ2:→Δshare↓ΔE2
 →(0≠EC←Δropen FF)ρ0
⍝
Δ3:Cs←2,5,0ρDs←'     Reading ',(¯2↓FF),' ...',0ρCs←7,5,0ρDs←⍙C[2;18]
 →(0≠↑(EC ⍙MR)←Δreadv)ρΔE1
 →(0≠↑(EC ⍙MC)←Δreadv)ρΔE1
 ⍙MA←MAT←(0,⍙MC)ρ⍳0
Δ31:→(0≠↑(EC MM)←Δreadv)ρΔE1
 MAT←MAT,[1]MM
 →(0≠↑(EC MM)←Δreadv)ρΔE1
 →(1≤⍙MR←⍙MR-1)ρΔ31,⍙MA←⍙MA,[1]MM
 →(0≠↑(EC ⍙MP)←Δreadv)ρΔE1
 →(0≠↑(EC ⍙LR)←Δreadv)ρΔE1
 →(0≠↑(EC ⍙LC)←Δreadv)ρΔE1
 →(0≠↑(EC ⍙LS)←Δreadv)ρΔE1
 →(0≠↑(EC F)←Δreadv)ρΔE1
 →(EC≠0)ρΔE1
 Δclose
 '→ΔE1' ⎕EA '(⍙UC ⍙UR ⍙AR ⍙AC ⍙RO ⍙CO ⍙OR ⍙OC MARKR MARKC HIGHR HIGHC)←F'
 (⍙WR ⍙WC)←(⍙SR-3+⍙UR)(1+⍙SC-⍙UC)
 ⍎(MARKR>0)/'⍙UL[1]←''M'''
 ⍎(HIGHR>0)/'⍙UL[3]←''S'''
 Cs←7,5,0ρDs←⍙C[1;18]
 →0,RC←1
⍝
ΔE2:→0,ρCs←11,1,0ρCs←2,5,0ρDs←'AP210 not active'
ΔE1:Δclose
 →Δ1,ρCs←11,1,0ρCs←2,5,0ρDs←'AP210: ',⍙ap210[|EC;]
```

```
[ 0] RC←Recode;UNI;VAL;D;I;COL
[ 1] ⍝
[ 2] ⍝  Replace elements in current column by new ones of character type
[ 3] ⍝  RC:  0 - no changes,  1 - screen update necessary
[ 4] ⍝
[ 5]  Cs←2,5,0⍴Ds←'      Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 6]  RC←↑0,I←⍴UNI←,((D⍳D)=⍳⍴D)/D←COL←⍕¨MAT[;⍙AC]
[ 7] Δ1:→(27=1↑VAL←VAL Input 'Enter new value for ',VAL←⍕,I⊃UNI)⍴Δ2
[ 8]  →(1≤I←I-RC←1)⍴Δ1,⍴COL[(D≡¨UNI[I])/⍳⍴D]←⊂VAL
[ 9] ⍝
[ 10] Δ2:→RC↓0
[ 11]  MAT[;⍙AC]←COL
[ 12]  ⍙MP[3 4 5;⍙AC]←0,'C',99
[ 13]  I←PlaceEntry ⍙AC
[ 14]  CFLAG←1
```

```
[ 0] Refresh;C;COL;LAB;W;N
[ 1] ⍝
[ 2] ⍝  Generate screen display matrix, column labels, column types
[ 3] ⍝
[ 4]  (⍙MR ⍙MC)←⍴MAT
[ 5]  C←↑1,⍴⍙CL←2 0⍴⍙DM←(⍙MR,0)⍴⍙CT←⍳0
[ 6] Δ1:W←DisCol                                    ⍝ get column width
[ 7]  ⍙DM←⍙DM,[2]W↑[2]COL                           ⍝ append column
[ 8]  ⍙CL←⍙CL,(2,W)⍴(W↑⍕C),W↑⊃⍙LR[C]
[ 9]  ⍙CT←⍙CT,W↑(,⍙MP[4;C]),(⍕W-2),,⊃(⍙MP[5;C]≠99)/⊂'.',⍕⍙MP[5;C]
[ 10]  →(⍙MC≥C←C+1)⍴Δ1
[ 11]  ⍙MP[1;]←0,¯1↓+\2+⍙MP[2;]                     ⍝ start posistion of columns
[ 12]  ⍙DA←(¯1↓2+⍙MP[2;])/⎕AF ⍙C[1;1+⍙MA]          ⍝ expanded color attributes
```

```
[ 0] NVEC←SUB Replace OVEC
[ 1] ⍝
[ 2] ⍝  Replace all 1↑SUB in OVEC by 1↓SUB giving NVEC
[ 3] ⍝
[ 4]  OVEC[(OVEC=1↑SUB)/⍳⍴OVEC]←⊂1↓SUB
[ 5]  NVEC←∊OVEC
```

```
[ 0] RC←ResPrt
[ 1] ⍝
[ 2] ⍝  Reset printer parameters and menu to default values  (RC always 0)
[ 3] ⍝
[ 4]  ⍙pvar←PRINT
[ 5] ⍝
[ 6]  ⍙menu3[5;]←30↑' F5   Start ',((1+⍙pvar[3])⊃'Landscape' 'Portrait'),' mode'
[ 7]  ⍙menu3[6;27+⍳3]←'550'⍕⍙pvar[4]
[ 8]  ⍙menu3[7;27+⍳3]←'550'⍕⍙pvar[5]
[ 9]  ⍙menu3[8;27+⍳3]←'550'⍕⍙pvar[6]
[ 10]  ⍙menu3[9;27+⍳3]←'550'⍕⍙pvar[7]
[ 11]  RC←0
[  0] RC←Rotate;R;C;RATT
[  1] ⍝
[  2] ⍝  Rotate matrix or marked area if anything marked
[  3] ⍝  RC: 1-changes, 0-no changes, set (with CFLAG) in BackLabel
[  4] ⍝
[  5]  Cs←2,5,0⍴Ds←'     Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[  6]  ⍎((⍳RC←0)≡R←(∨/⍙MA≥8)/⍳⍙MR)/'→Δ1,⍴(R C)←⍳¨⍴⍙MA'   ⍝ Find row/col numbers
[  7]  C←(∨⌿⍙MA≥8)/⍳⍙MC                                  ⍝ of marked area
[  8] Δ1:(ROMAT RATT)←⍉¨R AddLabel C                      ⍝ Create rotated submat
[  9]  ⍎(USER 'ROMAT')/'RC←(R C)BackLabel⍉ROMAT'          ⍝ Implement changes
```

```
[  0] SHOW V;ΔE
[  1] ⍝
[  2] ⍝  Show the variable V in the APL2 environment, return to UEDIT
[  3] ⍝  by hitting return.
[  4] ⍝
[  5]  ⍎(1=1⊃ΔE←⎕EC '⎕←V')/3
[  6]  V←((0=1⊃ΔE)/'Could not execute'),(0≠1⊃ΔE)/'Successfully executed'
[  7]  →(0≠1⊃ΔE)/Δ1
[  8]  3⊃ΔE,0⍴Cfs←11 1
[  9] Δ1:⍞,⎕←⎕TC[2],'Press Enter to Proceed'
```

```
[  0] RC←SXout;FILE;OUTMAT;COL;N
[  1] ⍝
[  2] ⍝  Export matrix columns to StatXact;  RC: always 0
[  3] ⍝
[  4]  COL←''
[  5] Δ1:→(27=1↑COL←COL Input 'Enter column numbers to export')⍴RC←0
[  6]  ⍎(COL≡,' ')/'COL←⍕⍳⍙MC'
[  7]  '→ΔE1' ⎕EA 'COL←⍎COL'                       ⍝ must be numeric
[  8]  →(∧/COL∊⍳¯1↑⍴MAT)↓ΔE3                        ⍝ columns must exist
[  9] ⍝
[ 10]  FILE←''
[ 11]  →(27=1↑FILE←GetFileName FILE)⍴0
[ 12]  Cs←2,5,0⍴Ds←'     Preparing ',FILE,' ...'
[ 13]  OUTMAT←⍕(⍳⍙MR),⊃MAT[;COL]
[ 14]  N[('¯'=N)/⍳⍴N←,OUTMAT]←'-'
[ 15]  OUTMAT←(⍴OUTMAT)⍴N
[ 16] ⍝
[ 17]  Cs←2,5,0⍴Ds←'     Writing ',FILE,' ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 18]  ⎕WA←(⊂[2]OUTMAT)Δfv FILE
[ 19]  →0
[ 20] ⍝  Error handling
[ 21] ΔE1:→ΔE0,MSG←'Input must be a numeric vector'
[ 22] ΔE3:MSG←'Non existing column number specified'
[ 23] ΔE0:→Δ1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG
[  0] RC←SaveMat F;FF;N;SV;EC
[  1] ⍝
[  2] ⍝  Save matrix and its parameters;  RC: 1 - ok,  0 - escape/error
[  3] ⍝
[  4]  RC←0
[  5]  →((8<⍴F)∨2=⎕NC 'RATT')⍴∆1
[  6]  ⍎(~F≡'')/'→∆2,⍴FF←PATH,F,''.UED,A'''
[  7] ∆1:→(27=1↑F←(Dtb 8↑F)Input 'Enter file name')⍴0
[  8]  →(F≡,' ')⍴∆1
[  9]  FF←PATH,((∧\F≠'.')/F),'.UED,A'
[ 10]  →∆share↓∆E2
[ 11]  EC←∆ropen FF
[ 12]  ∆close
[ 13]  ⍎(EC=0)/'→(GetYN F,'' already exists, delete it'')↓∆1'
[ 14] ⍝
[ 15]  Title F
[ 16] ⍝
[ 17] ∆2:Cs←2,5,0⍴Ds←'     Writing ',(¯2↓FF),' ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
[ 18]  →∆share↓∆E2
[ 19]  →(∧/0 2≠EC←∆delete FF)⍴∆E1
[ 20]  →(0≠EC←∆wopen FF)⍴∆E1
[ 21]  →(0≠EC←∆writev ⍙MR)⍴∆E1
[ 22]  →(0≠EC←∆writev ⍙MC)⍴∆E1,N←1
[ 23] ∆21:→(0≠EC←∆writev MAT[N;])⍴∆E1
[ 24]  →(0≠EC←∆writev ⍙MA[N;])⍴∆E1
[ 25]  →(⍙MR≥N←N+1)⍴∆21
[ 26]  →(0≠EC←∆writev ⍙MP)⍴∆E1
[ 27]  →(0≠EC←∆writev ⍙LR)⍴∆E1
[ 28]  →(0≠EC←∆writev ⍙LC)⍴∆E1
[ 29]  →(0≠EC←∆writev ⍙LS)⍴∆E1
[ 30]  N←⍙UC ⍙UR ⍙AR ⍙AC ⍙RO ⍙CO ⍙OR ⍙OC MARKR MARKC HIGHR HIGHC
[ 31]  →(0≠EC←∆writev N)⍴∆E1
[ 32]  ∆close
[ 33] ⍝
[ 34]  ⍎(ORIG←F),'←CompMat MAT'
[ 35]  ⍎(2=⎕NC 'RATT')/'RATT←⍙MA'
[ 36]  Cs←7,5,0⍴Ds←⍙C[1;18]
[ 37]  →0,RC←1
[ 38] ⍝
[ 39] ∆E1:→0,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←'AP210: ',⍙ap210[|EC;]
[ 40] ∆E2:Cs←11,1,0⍴Cs←2,5,0⍴Ds←'AP211 not active - saving to workspace only'
```

```
[  0] SetPtr
[  1] ⍝
[  2] ⍝  Set pointers for a new matrix
[  3] ⍝
[  4]  ⍙AR←⍙AC←⍙RO←⍙CO←1          ⍝ coord of cursor cell/upper left window cell
[  5]  MARKR←MARKC←HIGHR←HIGHC←0       ⍝ first corner of block to mark/highlight
[  6]  ⍙OR←⍙OC←0                                ⍝ flags: screen update necessary
[  7]  (⍙WR ⍙WC)←(⍙SR-3+⍙UR)(1+⍙SC-⍙UC)         ⍝ size of edit window
[  8]  ⍙DA←(¯1↓2+⍙MP[2;])/⎕AF ⍙C[1;1+⍙MA]       ⍝ expanded color attributes
```

```
RC←Shadow;R;C;COLR;RATT;⍙SP
⍝
⍝  Shadow un-highlighted areas;  RC: 1-done, 0-escape/no changes
⍝  Sets CFLAG in BackLabel if necessary
⍝
 →(0=COLR←GetColor)⍴RC←0                            ⍝ Escape pressed
 →((⍳0)≡R←(∨/⍙MA≥COLR)/⍳⍙MR)⍴0                       ⍝ Quit if no highlights
 Cs←2,5,0⍴Ds←'     Working ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
 (SHMAT RATT)←R AddLabel C←(∨⌿⍙MA≥COLR)/⍳⍙MC         ⍝ Create submatrices
 ⍎(USER 'SHMAT')/'RC←(R C)BackLabel SHMAT'           ⍝ Implement changes
```

```
ShowCell;C
⍝
⍝  Set color attributes of current cell to 'active'
⍝
 Ds←(⍙UR+⍙AR-⍙RO),⍙UC+⍙MP[1;⍙AC]-⍙MP[1;⍙OC←⍙CO]
 C←⍙C[2;1+⍎(1+∨/(⍙AC,⍙AR)>⍙MC,⍙MR)⊃'⍙MA[⍙AR;⍙AC]' '0']
 Cs←1,9,0⍴Ds←Ds,1,((1+⍙SC-Ds[2])⌊⍙MP[2;⍙AC]),0,C
```

```
RC←Sort;Label;VEC;MSG;S
⍝
⍝  Sort rows on different columns; RC: 1 - sorted,  0 - escape
⍝
 VEC←''
Δ1:VEC←VEC Input 'Enter col×'' s in major-minor order ( >0 ⍋ , <0 ⍒ )'
 →(27=1↑VEC)⍴RC←0
⍝
 '→ΔE1' ⎕EA 'VEC←⍎VEC'                          ⍝ Input must be numeric
 →(∧/(|VEC)∊⍳⍙MC)↓ΔE2                           ⍝ Column numbers must exist
⍝
 Cs←2,5,0⍴Ds←'     Sorting ...',0⍴Cs←7,5,0⍴Ds←⍙C[2;18]
Δ2:S←MAT sortΔsub ¯1↑VEC
 (MAT ⍙MA ⍙DM ⍙DA ⍙LC)←(MAT[S;])(⍙MA[S;])(⍙DM[S;])(⍙DA[S;])(⍙LC[S])
 →(1≤⍴VEC←¯1↓VEC)⍴Δ2
⍝
 →(3>⍴⍙LR)⍴Δ3
 →(⍙LR[3]≡⊂'Cum.')↓Δ3                           ⍝ special handling if within
 VEC←(MAT[;3]≠MISSΔN)/⍳⍙MR                      ⍝   freq tabs: recalculate
 MAT[VEC;3]←+\MAT[VEC;1]                        ⍝   cumulative frequencies
 ⎕WA←PlaceEntry 3
⍝
Δ3:DISL←VerShift
 →0,CFLAG←RC←1
⍝
⍝  Error handling
ΔE1:→ΔE0,MSG←'Input must be a numeric vector'
ΔE2:MSG←'Non existing column numbers specified'
ΔE0:→Δ1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG
```

```
RC←StatParam
⍝
⍝  Generate matrix parameters for a new, overlay matrix; takes the
⍝  information from ⍙SP;  RC: always 1
⍝
 MAT←⍎ORIG
 (⍙MA ⍙LC ⍙LR ⍙LS ⍙UR RC)←(RATT)(1↓MAT[;1])(1↓MAT[1;])(' ')3 1
 (⍙MR ⍙MC)←⍴MAT←1 1↓MAT
 ⍙UC←7+1↓⍴⊃⍙LC
 ⍎(∨/,⍙MA>0)/'⍙UL[3]←''H'''
 ⍙MP←(5,1↓⍴⍙SP)⍴0
 ⍙MP[3 5;]←⍙SP
 ⍙MP[4;]←('CND')[1+⍙MP[3;]]
 ⍙MP←⍙MP,0,3,0,0,0
 SetPtr
```

```
Title F;N;MSG
⍝
⍝  Create and display status line
⍝
 N←⌊0.5×⍙SC-21+⍴MSG←F,'  [',(⍕↑⍴MAT),'×',(⍕¯1↑⍴MAT),']'
 Cs←4,4,0⍴Ds←' UEdit 1.00',(N↑' '),MSG,(N↑' '),'F1 - Help '
```

```
RC←ToDays D;DD;MM;YY
⍝
⍝  Validate Date D and convert to days since February 29, 0000
⍝  D:  Character vector of input data 'MM DD YYyy' or different
⍝      order depending on the global variable DATE
⍝  RC: MISS∆N  if D is blank
⍝      0        if D is invalid
⍝      days since Feb 29, 0000 otherwise
⍝
 →(∧/D=' ')⍴0,RC←MISS∆N
 RC←0
 D[(D∊'-/.')/⍳⍴D←,D]←','
⍝  '→0' ⎕EA '→((⍴D←,⍎D)∊2 3)↓0'
 '→0' ⎕EA '→((⍴D←⍎D)∊2 3)↓0'
 ⍎(2=⍴D)/'D←(DATE⍳1)⊃(⎕TS[1],D)(D[1],⎕TS[1],D[2])(D,⎕TS[1])'
 (YY MM DD)←D[DATE⍳⍳3]
 YY←YY+(YY<100)×100×⌊⎕TS[1]÷100
 →(MM∊⍳12)↓0                                           ⍝ month ≤ 12
 →(DD∊⍳(31 29 31 30 31 30 31 31 30 31 30 31)[MM])↓0    ⍝ days/month
 ⍎(∧/2 29=MM DD)/'→((0≠4|YY)∨(0=100|YY)∧0≠400|YY)⍴0'   ⍝ leap year
⍝
 RC←(365×YY)+-/⌊(YY←YY-MM≤2)∘.÷4 100 400               ⍝ convert date
 RC←RC+DD+(306 337 0 31 61 92 122 153 184 214 245 275)[MM]
```

```
RC←ToMarg;L;V
⍝
⍝  Set topt margin
⍝
 L←⍕⍙pvar[6]
Δ1:→(27=1↑L←L Input 'Enter new top margin')ρRC←0
 '→ΔEr' ⎕EA '⍙pvar[6]←V←999⌊⌈(.2×⍙pvar[7])⌈|⍎L'
 →0,ρ⍙menu3[8;27+⍳3]←'550'⍕V
⍝
ΔEr:→Δ1,0ρCs←11,1,0ρCs←2,5,0ρDs←'Input must be numeric'
```

```
RC←NT ToggType C;TYP;DEC;K;I
⍝
⍝  Change type/format of column C (to T if T given)
⍝  RC:  1 - screen update necessary,  0 - not necessary
⍝
 →(⍙AC>⍙MC)ρRC←0
 →(2=⎕NC 'NT')ρΔ2
 NT←''
Δ1:→(27=1↑NT←NT Input 'Enter new format ( C / D / Nx / Ex / A )')ρ0
Δ2:→(ParseType NT)↓ΔE1
 Cs←2,5,0ρDs←'     Converting ...',0ρCs←7,5,0ρDs←⍙C[2;18]
 →(1+⍙MP[3;C],TYP)⊃(0,Δ3,Δ4)(Δ5,Δ6,Δ8)(Δ7,Δ8,0)
⍝
Δ3:K[I←(∧/¨K=' ')/⍳ρK←MAT[;C]]←⊂⍕MISSΔN               ⍝ char -> num
 '→ΔE1' ⎕EA '→Δ8,MAT[;C]←⍎¨K'
Δ4:→(∨/0=K←∊ToDays¨MAT[;C])ρΔE1                       ⍝ char -> date
 →Δ8,MAT[;C]←K
⍝
Δ5:→Δ8,ρMAT[;C]←Dtb¨⊂[2]NumToChar MAT[;C]             ⍝ num -> char
Δ6:→(DEC=⍙MP[5;C])↓Δ8,0                               ⍝ num -> date
⍝
Δ7:MAT[;C]←Dtb¨⊂[2]FromDays MAT[;C]                   ⍝ date -> char
⍝
Δ8:⍙MP[3 4 5;C]←TYP,('CND')[TYP+CFLAG←1],DEC          ⍝ Update parameters
 ⎕WA←PlaceEntry C                                     ⍝   and display
 →0,RC←1
⍝
⍝  Error handling
ΔE1:→Δ1,0ρCs←11,1,0ρCs←2,5,0ρDs←'Invalid format'
```

```
TryNum I;N
⍝
⍝  Try to make column N of 'Get' matrix numeric; if that doesn't
⍝  work convert matrix column I to character type
⍝
 N←I-⍙AC-1
 '' ⎕EA '→0,NMAT[;N]←⍎¨NMAT[;N]'
 ⎕WA←'C' ToggType I
```

```
[  0] R←VerShift;I
[  1] ⍝
[  2] ⍝  Adjust line numbers, row labels, returns DISL
[  3] ⍝
[  4]  R←(⍙WR,5)⍴'2220 '⍕I←¯1+(⍙OR←⍙RO)+⍳⍙WR                ⍝ column numbers
[  5]  ⍎(⍙UC>6)/'R←(R,⍙WR↑[1](⊃⍙LC)[(I≤⍙MR)/I;]),'' '''     ⍝ column labels
[  6]  Cs←1,2,0⍴Ds←⍙UR,1,⍙WR,(⍙UC-1),2,⍙C[1;16]              ⍝ re-define field
[  7]  Cs←4,2,0⍴Ds←R←,R                                      ⍝ display field


[  0] C←ask∆col PROMPT;C;MSG
[  1] ⍝
[  2] ⍝  Get column number to work on, return immediately if escape or ' '
[  3] ⍝
[  4]  C←''
[  5] ∆1:→(27=1↑C←C Input PROMPT)⍴0
[  6]  →(C≡,' ')⍴0
[  7]  '→∆E1' ⎕EA '→(0≠⍴⍴C←⍎C)⍴∆E2'
[  8]  →(C∊⍳⍙MC)⍴0
[  9]  →∆E0,MSG←'Non existing column number specified'
[ 10] ∆E1:→∆E0,MSG←'Input must be a numeric vector'
[ 11] ∆E2:MSG←'Specify only 1 column, please'
[ 12] ∆E0:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG


[  0] P←ask∆par C
[  1] ⍝
[  2] ⍝  Get classification data for numeric vector:
[  3] ⍝     lower limit, upper limit, number of classes
[  4] ⍝
[  5]  →(⍙MP[3;C]=0)⍴P←0                    ⍝ no classification for character columns
[  6]  P←''
[  7] ∆1:P←P Input 'Column ',(⍕C),': lower limit, upper limit, number of classes'
[  8]  →(27=1↑P)⍴0
[  9]  →(1=⍙MP[3;C])⍴∆2
[ 10]  →(3≠⍴P←Parse P)⍴∆E1
[ 11]  '→∆E2' ⎕EA '→(∨/0=P←(∊ToDays¨2↑P),⍎⊃P[3])↓0,∆E3'
[ 12] ∆2:'→∆E2' ⎕EA '→(3=⍴P←⍎P)⍴0'               ⍝ data must be numeric, 3 elements
[ 13] ⍝
[ 14] ∆E1:→∆E0,MSG←'Specify a 3-element vector, please'
[ 15] ∆E2:→∆E0,MSG←'Invalid number specified'
[ 16] ∆E3:MSG←'Invalid date specified'
[ 17] ∆E0:→∆1,0⍴Cs←11,1,0⍴Cs←2,5,0⍴Ds←MSG


[  0] RC←ff
[  1] ⍝
[  2] ⍝  Print form feed
[  3] ⍝
[  4]  RC←∆print ⎕AF 13 12
```

```
R←VAL freqΔcnt CLASS;A;B
⍝
⍝  Counts ocurrences of elements of ⍳CLASS in VAL. VAL must be
⍝  sorted (⍋ or ⍒) and must not contain elements not in ⍳CLASS.
⍝
 R←CLASSρ0
 R[A/VAL]←B-(ρB)ρ0,B←(A←(¯1↓VAL≠1⌽VAL),1)/⍳ρVAL
```

```
R←Z sortΔsub COL;C;S
⍝
⍝  Sort rows of matrix Z on column COL.  Sort ⍋ if COL>0, ⍒ if COL<0.
⍝  The return vector R contains the index order of a sorted matrix Z,
⍝  the matrix itself is NOT sorted.  The sorting is case insensitive,
⍝  i.e., lower and upper case characters are considered equal.
⍝
 S←2 64ρ' !⍝×$÷∧''()⋆+,-./0123456789:;<=>?ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
   ∧_ !⍝×$÷∧''()⋆+,-./0123456789:;<=>?abcdefghijklmnopqrstuvwxyz[\]∧_'
 →(1+(COL<0)+2×⍙MP[3;C←|COL]>0)⊃ChrPos,ChrNeg,NumPos,NumNeg
ChrPos:→0,R←S⍋⊃Z[;C]
ChrNeg:→0,R←S⍒⊃Z[;C]
NumPos:→0,R←⍋⊃Z[;C]
NumNeg:R←⍒⊃Z[;C]
```

```
Δclose
 ⎕WA←⎕SVR¨'Cz' 'Dz'        ⍝ Close file, retract Cz, Dz
```

```
RC←Δdelete F
 Cz←'DL,',F           ⍝ Delete file F
 RC←Cz                ⍝ RC: AP210 return code
```

```
R←A Δfv B;⎕IO;Cz;Dz;E;F
⍝
⍝  Emulation of APL2/370 ΔFV built-in function using AP210
⍝  modified to speed up UEdit (original in IBM's FILES workspace)
⍝
 →Δshare↓ΔE1
 →((⎕IO←R←0)=⎕NC 'A')ρΔ3
⍝
⍝  Write
⍝
 →(∧/0 2≠E←Δdelete B)ρΔE2                   ⍝ Delete existing file if any
 →(0≠E←Δwopen B,',D')ρΔE2                   ⍝ Open new file for writing
 F←(B←0),2+⌈/∊ρ¨A                          ⍝ Max. record length
Δ1:→(B≥ρA)ρ0
 Dz←((-(⌽' '=E)⍳0)↓E←B⊃A),⎕TC[1 2]         ⍝ Write each record in turn
 Cz←5,F
 →F←(0≠1↑E←Cz)ρΔE2
 →Δ1,B←B+1
⍝
⍝  Read
⍝
```

```
[ 21] Δ3:→(0≠↑E←Δropen B,',D')ρΔE2          ⍝ Open file for read
[ 22]  R←''                                 ⍝ Init result variable
[ 23]  B←0×F←128                            ⍝ Init counter ×records, scan length
[ 24] Δ4:Cz←4
[ 25] Δ5:→((¯45 ¯44=↑E),0≠↑E←Cz)/Δ6,Δ7,ΔE2
[ 26]  R←R,⊂(-(⌽' '=E)⍳0)↓E←¯2↓Dz                 ⍝ Read each record in turn
[ 27]  →Δ4,B←B+1
[ 28] Δ6:→0,ρR←(-(⎕AF 26)=↑↑¯1↑R)↓R              ⍝ Remove EOF record (if there)
[ 29] Δ7:→(65400=F)ρΔE3
[ 30]  →Δ5,Cz←4,B,F←65400⌊2×F                     ⍝ Double scan length
[ 31] ⍝
[ 32] ΔE1:→0,ρCs←11,1,0ρCs←2,5,0ρDs←'AP210 not active'
[ 33] ΔE2:→0,ρCs←11,1,0ρCs←2,5,0ρDs←'AP210 error: ',⍙ap210[¯1+|E;]
[ 34] ΔE3:Cs←11,1,0ρCs←2,5,0ρDs←'Invalid file'


[  0] RC←Δprint V;X
[  1] ⍝
[  2] ⍝  Send output stream to AP81
[  3] ⍝  RC: 0 - OK,   1 - AP81 missing
[  4] ⍝
[  5]  ⎕WA←81 ⎕SVO 'X'
[  6]  →(2≠⎕SVO 'X')ρΔErr
[  7]  X←⍕V
[  8]  ⎕WA←⎕SVR 'X'
[  9]  →RC←0
[ 10] ⍝
[ 11] ΔErr:Cs←11,1,0ρCs←2,5,0ρDs←'AP81 is not active'
[ 12]  RC←1


[  0] Z←Δreadv;RC
[  1]  Cz←4                         ⍝ Read record
[  2]  ⍎((RC←Cz)∊¯44 ¯46)/'EC←0'    ⍝ from variable
[  3]  Z←RC Dz                      ⍝ length file

[  0] RC←Δropen F
[  1]  Cz←'IR,',F                   ⍝ Open file for Read Only
[  2]  RC←Cz                        ⍝ RC: AP210 return code

[  0] RC←Δshare
[  1]  RC←210 ⎕SVO¨'Cz' 'Dz'        ⍝ Share global variables Cz, Dz with AP210
[  2]  RC←2=⎕SVO 'Cz'               ⍝ RC: 1 - ok,  0 - error

[  0] RC←Δwopen F
[  1]  Cz←'IW,',F                   ⍝ Open file for Read/Write
[  2]  RC←Cz                        ⍝ RC: AP210 return code

[  0] RC←Δwritev A
[  1]  Cz←5,0ρDz←A                  ⍝ Write record A to
[  2]  RC←Cz                        ⍝ variable length file
```

LIST OF REFERENCES

[1] Kendall, M. G., and Stuart, A., *The Advanced Theory of Statistics*, Vol. 2, Charles Griffin & Company Ltd., 1961

[2] Keith, I. A., *An APL Workspace for Analyzing Categorical Data with Application to the PERSEREC Database of Recent American Espionage*, Master's Thesis, Naval Postgraduate School, California, June 1989

[3] Bergquist, G. A., *APL Advanced Techniques and Utilities*, Zark, Inc., 1987

[4] Koucheravy, E. R., *An Analysis of Security Background Investigation Data and the Relationship With Subsequent Discharge*, Master's Thesis, Naval Postgraduate School, California, September 1988

[5] Conover, W. J., *Practical Nonparametric Statistics*, John Wiley & Sons, Inc., 1971

[6] Goodman, L. A., and Kruskal, W. H., *Measures of Association for Cross Classification*, Springer-Verlag, 1979

[7] Anscombe, F. J., *Computing in Statistical Science through APL*, Springer-Verlag, 1981

[8] Cochran, W. G., "The χ^2 Test of Goodness of Fit", *Annals of Mathematical Statistics*, Vol. 23, pp. 315–345, September 1952

[9] Roscoe, J. T., and Byars, J. A., "An Investigation of the Restraints with Respect to Sample Size Commonly Imposed on the Use of the Chi-Square Statistic", *Journal of the American Statistical Association*, Vol. 68, December 1973

[10] Chambers, J. M., Cleveland, W. S., Kleiner, B. Tukey, P. A., *Graphical Methods for Data Analysis*, Wadsworth & Brooks/Cole Publishing Company, 1983

BIBLIOGRAPHY

APL2 Programming: Language Reference, 2nd ed., International Business Machines Corporation, 1985

APL2 Programming: APL/2 for the IBM PC, International Business Machines Corporation, 1989

Brown, J. A., Pakin, S., Polivka, R. P., *APL2 at a Glance*, Prentice Hall, Inc., 1988

Gilman, L., and Rose, A. J., *APL: An Interactive Approach*, John Wiley & Sons, Inc., 1984

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5002

3. Federal Republic of Germany 1
Federal Department of Defense, P III 6
c/o German Military Representative USA/Canada
4000 Brandywine Street, N.W.
Washington, District of Columbia 20016-1887

4. Professor Peter A. W. Lewis, Code 55Lw 11
Naval Postgraduate School
Monterey, California 93943-5000

5. Professor Laura D. Johnson, Code 55Jo 1
Naval Postgraduate School
Monterey, California 93943-5000

6. Professor Harold J. Larson, Code 55La 1
Naval Postgraduate School
Monterey, California 93943-5000

7. Professor Robert R. Read, Code 55Re 1
Naval Postgraduate School
Monterey, California 93943-5000

8. LCDR Al Callahan, Code 55Ca 1
Naval Postgraduate School
Monterey, California 93943-5000

9. Professor Toke Jayachandaran, Code 53Ja 1
Naval Postgraduate School
Monterey, California 93943-5000

10. Defense Personnel Security Research and Education Center 3
(PERSEREC), Code 65
Naval Postgraduate School
Monterey, California 93943-5000

11. Dr. Ralph Carney 1
PERSEREC, Code 65
Naval Postgraduate School
Monterey, California 93943-5000

12. Dr. Roger Denk 1
PERSEREC, Code 65
Naval Postgraduate School
Monterey, California 93943-5000

13. Dr. Al Blum 1
IBM Research Center
P.O. Box 204
Yorktown Heights, New York 10598

14. Dr. Peter Welch 1
IBM Research Center
P.O. Box 204
Yorktown Heights, New York 10598

15. Dr. James Brown 1
IBM Santa Theresa Labs, J98 B252
555 Bailey Avenue
San Jose, California 95141

16. Norman Thomson 1
Mail Point 188
IBM UK Labs Ltd.
Hursley House, Hursley Park
Winchester, Hampshire S021 2JN
ENGLAND

619-588